

Capítulo 6

Conclusiones y Resultados

6.1. Conclusiones

Tras la realización de este proyecto se ha llegado a las siguientes conclusiones:

- La robótica es un elemento cada vez más presente en la vida cotidiana de las personas. Proyectos de investigación como URUS vienen demostrando que la robótica puede ser integrada en la vida diaria de las personas como una potente herramienta que flexibilice y facilite la vida de las mismas. Incluso en el mercado actual tenemos ejemplos de integración de robots en nuestro entorno, como el caso del famoso robot limpiador Roomba, de la empresa iRobot, que actualmente comercializa la tercera generación de su producto con un gran éxito comercial. La integración de la robótica en los entornos urbanos más allá de los límites de nuestro hogar será una realidad en los años venideros.
- Se ha comprobado la viabilidad del uso de las librerías gráficas Qt como una herramienta de desarrollo potente y versátil para la realización de interfaces de usuario. Aprovechando la cantidad de herramientas de desarrollo explicadas en el capítulo 3 la realización de interfaces de usuario, en cualquier grado de complejidad, es una tarea que se ve ampliamente facilitada al usar Qt.
- La utilización de la librería YARP como middleware para comunicaciones ha supuesto un gran acierto, ya que se ha consolidado como una poderosa herramienta que libera de gran cantidad de trabajo para el desarrollo de otros aspectos del robot, ofreciendo una abstracción de las comunicaciones en el concepto de puerto, muy fácil de implementar y de integrar en el resto del sistema. Se puede entender YARP como el *sistema circulatorio* del robot.

- La utilización de la herramienta LogPlayer ha resultado de gran utilidad para la simulación y el refinamiento de los algoritmos diseñados en la fase final del proyecto URUS.
- Personalmente, haber formado parte del equipo de desarrollo de URUS ha sido una gran experiencia formativa. Me ha permitido adquirir y desarrollar conocimientos en desarrollo en C++ bajo entornos linux, integración de librerías de desarrollo diversas, conocimientos hardware sobre sensores, desarrollo de drivers, etc., todo ello de la mano de unos excelentes profesionales y personas.

6.2. Futuras líneas de desarrollo

La interfaz gráfica Romeo HMI ha sido diseñada y realizada pensándose como una herramienta para el manejo del robot, pero orientada siempre para los desarrolladores del sistema. Por tanto, podemos pensar en dos vías de desarrollo bastante claras a partir de este proyecto. En primer lugar mejoras que no se llegaron a implementar en la interfaz actual o solucionar o mejorar problemas que aparecieron en el desarrollo de la misma y por otro lado la realización de una interfaz gráfica orientada para usuarios finales.

6.2.1. Mejoras en la interfaz gráfica actual

- *Integración de Qt y openCV.* Uno de los mayores problemas encontrados fue la integración entre Qt y openCV. La librería openCV, utilizada en módulos como sensormap o elevationmap, utiliza una serie de formatos gráficos de imagen propios, que deben ser convertidos a bitmaps para poder ser utilizados posteriormente por Qt. Esto deriva en un consumo excesivo de tiempo de procesado. Desarrollar una clase que implemente esta integración entre Qt y openCV facilitaría muchos trabajos amén de abrir nuevas posibilidades de visualización y de desarrollo.
- *Integración Google Maps.* En espacios donde el módulo GPS estuviese disponible sería muy interesante programar un widget que haciendo uso de la API de google maps nos permitiese aprovechar las funcionalidades de esta aplicación web, como por ejemplo mostrar la ubicación de Romeo-4R en una imagen satelital, cálculo de rutas, búsqueda de información en los alrededores del robot, etc.

6.2.2. Interfaz gráfica para usuarios finales

El siguiente paso lógico en la evolución del trabajo realizado en URUS por parte del GRVC consistiría en realizar desde cero una interfaz gráfica orientada a los

usuarios finales, donde se facilite la selección de las labores que pueda realizar el robot, taxi o seguimiento, ocultando al usuario información que realmente no necesita conocer, como la generación de los mapas de elevación, posiciones estimadas del robot, etc., mostrando un funcionamiento totalmente transparente al mismo.

6.3. Resultados obtenidos en Proyecto URUS

Como se comentó en la introducción, el trabajo realizado en este proyecto fin de carrera, se enmarca dentro de un proyecto de investigación Europeo, el proyecto URUS, por lo que para finalizar el contenido de esta memoria, recogemos algunos de los resultados más importantes que fueron obtenidos, por el equipo de trabajo del GRVC, dirigido por D. Aníbal Ollero Baturone y D. Luis Merino Cabañas, director de este proyecto fin de carrera y recogidos en [19] y [13].

Las áreas peatonales y urbanas, como por ejemplo un campus universitario, presentan una serie de retos para robots móviles tipo coche, tales como escaleras, rampas, pequeños bordillos, árboles, etc., a lo que debemos añadir personas en movimiento, por lo que la evitación de obstáculos dinámicos es un requerimiento esencial en este tipo de sistemas.

Por tanto, para una correcta recreación del entorno se hace necesario construir un modelo tridimensional del mismo, a partir del cual, el robot Romeo-4R es capaz de construir un mapa de transversabilidad (tal y como vimos en el apartado 2 de la memoria), en el que se indica para cada celda del mapa si ésta es transversable o no.

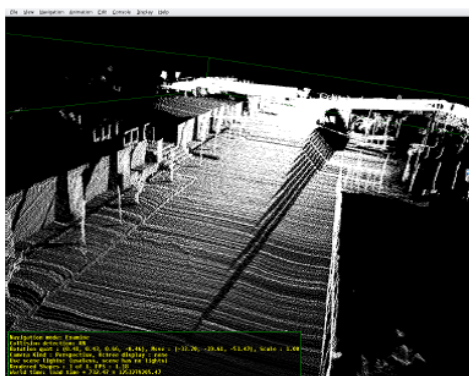


Figura 6.1: Proyecto URUS. Recreación entorno 3D.

Utilizando la videocámaras a bordo del robot Romeo-4R podemos obtener la posición de personas (lo que se puede emplear en tareas de guiado o de interacción con personas). Los algoritmos para determinar esta posición se basan en una combinación de detección y seguimiento.

El algoritmo de seguimiento se basa en la técnica mean-shift. Un algoritmo de detección de caras [26] se ejecuta en paralelo; los resultados de ambas técnicas se combinan, de modo que cuando el seguimiento se pierde, el detector permite recuperar a la persona.

Como resultado, el robot puede estimar la posición de la cara de una persona en el plano de la imagen. Aplicando razonamientos geométricos, sería posible estimar la posición de la persona con relación al robot.



Figura 6.2: Proyecto URUS. Reconocimiento facial.

Otro elemento de apoyo existente en el escenario de los experimentos, comentado en la introducción de este texto, es una red de cámaras IP fijas, las cuales pueden cubrir parte del entorno, y permiten el seguimiento de objetos de interés. Esta información puede combinarse con la propia información que obtienen los robots para mejorar la percepción del entorno.

El sistema empleado es capaz de seguir objetos de interés tanto dentro de una cámara como entre cámaras diferentes sin necesidad de una calibración explícita e incluso en ausencia de solape en sus campos de visión. La técnica ha sido desarrollada por Gilbert y Bowden [16].

La figura 6.3 muestra un ejemplo en el que la misma persona es seguida por 3 cámaras que tienen muy poca o ninguna superposición.

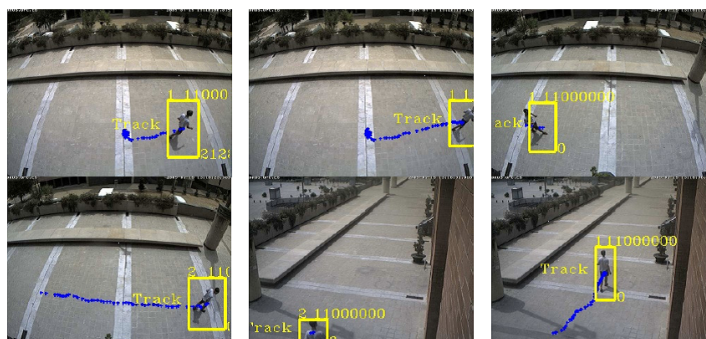


Figura 6.3: Proyecto URUS. Red cámaras IP - Identificación y seguimiento.

Para el guiado de una persona, el robot debe tener capacidad de navegación. La

figura 6.4 muestra algunos resultados de navegación de Romeo en el escenario del proyecto URUS. Uno de estos experimentos corresponde a una hipotética misión de guiado, en la que el robot debe guiar a una persona de un punto a otro del mapa.

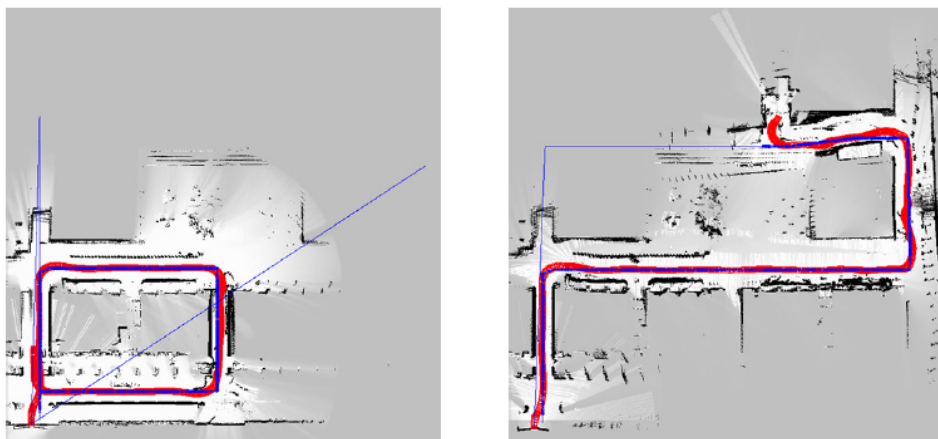


Figura 6.4: Proyecto URUS. Experimentos Navegación Romeo-4R.

Para poder guiar a alguien, es necesario estimar la posición de dicha persona para ajustar el movimiento del robot al de la persona. Tal y como se comentó igualmente en la introducción, para ayudar en las tareas de reconocimiento del entorno disponemos de una red de sensores desplegada por toda la zona de los experimentos, compuesta por decenas de nodos estáticos capaces de monitorizar diversas variables del entorno.

Al mismo tiempo, esta red se emplea para estimar la posición de un nodo móvil de la misma red a partir de la potencia de señal radio recibida por los nodos estáticos. Así podemos conseguir medidas adicionales para mejorar el seguimiento de personas, tal y como vemos en la figura 6.5.

El algoritmo para la estimación de la posición del nodo móvil está basado en un filtro de partículas. En dicho filtro, la estimación actual de la posición del nodo se describe por un conjunto de partículas, cada una de las cuales representa una hipótesis sobre la posición real de la persona que lleva el nodo. En cada iteración del filtro, modelos de movimiento de la persona e información del mapa se emplea para predecir la posición futura de las partículas. El uso de mapas permite descartar movimientos poco probables.

Cuando los nodos estáticos reciben nuevos mensajes del nodo móvil, el peso de las distintas partículas se ajusta teniendo en cuenta la potencia recibida.

Utilizando modelos de propagación de la señal radio, es posible calcular la



Figura 6.5: Proyecto URUS. Redes de Sensores (I).

verosimilitud de una hipótesis (partícula) a partir de la distancia al nodo receptor según dicha hipótesis [13].

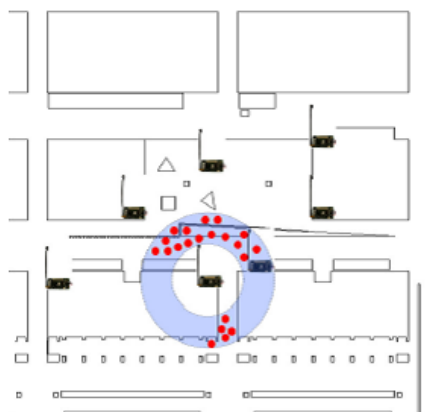


Figura 6.6: Proyecto URUS. Redes de Sensores (II).

Cada mensaje restringe la posición de las partículas a un anillo circular alrededor del nodo receptor en función de la potencia recibida, situación representada en la figura 6.6

Como resultado, el filtro proporciona estimaciones de la posición 3D del nodo móvil con una precisión que puede llegar a ser de 1 metro en función de la densidad de nodos de la red. La figura 6.7 muestra la evolución de las partículas para un experimento de guiado.

La figura 6.8 muestra la posición estimada, comparada con la posición del robot que la guía (que se encuentra unos metros más adelante).

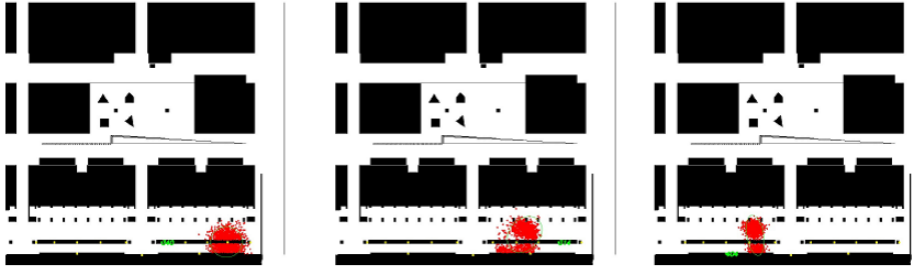


Figura 6.7: Proyecto URUS. Experimento de guiado (I).

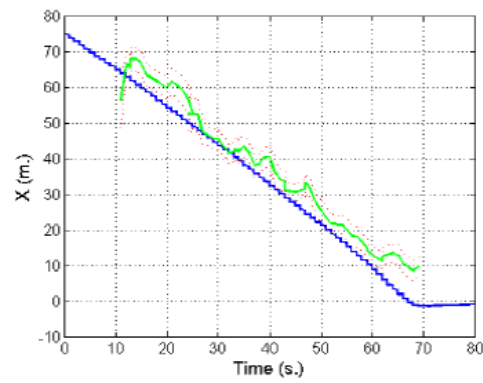


Figura 6.8: Proyecto URUS. Experimento de guiado (II).

6.4. Uso de las lecciones aprendidas en posteriores proyectos

La positiva experiencia en el desarrollo de interfaces gráficas basadas en Qt se ha visto plasmada en posteriores proyectos realizados por el *Grupo de Robótica, Visión y Control*, donde con los conocimientos adquiridos en el desarrollo de este proyecto fin de carrera, se han extendido las posibilidades de las librerías Qt, integrándolas en entornos multirobot y con otras librerías gráficas tales como Open Scene Graph (librería para gráficos 3D) y Open Street Map (librería para servicios de información geográfica), tal y como la Ground Control Station desarrollada para trabajar con múltiples UAVs en el entorno del Proyecto INTEGRA.



Figura 6.9: Qt Multi UAV Ground Control Station.