



R2-G2

ROBOTS MÓVILES

José Llorens M. y Asahel Hernández T.

UA | INGENIERÍA ROBÓTICA

INDICE

1.	Introducción y objetivos	2
2.	Detección e identificación de personas	2
3.	Detección de QRs y objetos.....	2
4.	Control Visual	2
5.	Exploración del entorno y mapeado	2
6.	Segmentación del entorno	2
7.	Navegación.....	3
8.	Patrullaje	4

1. Introducción y objetivos

2. Detección e identificación de personas

3. Detección de QRs y objetos

4. Control Visual

5. Exploración del entorno y mapeado

6. Segmentación del entorno

Para realizar una correcta gestión del entorno debemos poder distinguir entre las distintas habitaciones y pasillos, de forma que podamos indicarle al robot si debe ir o patrullar una determinada estancia.

Primeramente, se intentó emplear una librería que tenía varios métodos implementados. Sin embargo, no se consiguió configurar correctamente, por lo que se determinó que el mejor método para nuestra posibilidad era emplear métodos de transformación de la imagen.

Para ello se requiere el mapa .pgm del área y el yaml para poder calcular la equivalencia entre los píxeles y las coordenadas del mapa. Se requiere convertir la imagen a blanco y negro binarizado para posteriormente realizar un proceso de erosionar las áreas y luego dilatarlas.

#Código

Una vez hecho esto se obtiene las regiones se crea un mapa de colores que nos servirá para calcular los centroides de cada estancia, mediante una serie de fórmulas y considerando el punto 0,0 la posición del robot dada por el .yaml.

#Código

Luego se crea una máscara para cada una de las áreas y se superponen en el mapa pgm. Se guarda el mapa de cada área concreta de forma que podemos hacer que el robot navegue por el área específica que se desee.

#Código

Finalmente, se guarda las coordenadas de los centroides, el color asignado a esa área y la ruta al mapa del área segmentada.

#Código

Finalmente, se obtendrá los siguientes elementos que se usarán para la navegación y la patrulla.

Un mapa de las áreas segmentadas en color.

#Imagen

Un csv con los centroides, el color de las áreas y la ruta al mapa segmentada de cada área.

#Matriz de ejemplo con el csv

Y unos mapas de las areas segmentadas.

#Imagen

7. Navegación

Una funcionalidad implementada ha sido la de ser capaz de enviar ordenes de movimiento a coordenadas y orientaciones específicas al robot, mediante la librería Move.

#Código a mover_a_goal

Esto junto con la capacidad de poder calcular las coordenadas de un pixel determinado, nos permite que desde una interfaz pulsando sobre un pixel accesible del mapa podemos indicarle que se dirija a esa posición. Se ha de pulsar dos veces sobre el mapa, el primer click indica la posición de destino, el segundo indica la orientación que debe tener el robot cuando llegue.

#Código click_event_mapa,

El método empleado para la conversión entre pixeles y coordenadas es el mismo que se usó para para calcular las coordenadas de los centroides

Código convertir_a_coordenadas,

Para calcular la orientación se hace simplemente calculando el ángulo theta entre ambos puntos.

Código calcular_orientacion,

El resultado obtenido es el siguiente, siendo este satisfactorio pudiendo ir a cualquier punto que se desee.

#Video de la interfaz

8. Patrullaje

Para la tarea de realizar patrullas, dado el mapa de color y el csv con el color y la ruta al mapa segmentando se optado por otro mapa iterativo que al pulsar encima de un área coloreada se llame al script de patrulla. El robot solo patrullara sobre área que se ha seleccionado.

Para ello primeramente se detecta el color del pixel seleccionado y se compara con los colores guardados en csv.

#Codigo detectar_color_pixel

Si coincide se llama al script de patrulla, que envía la ruta al mapa segmentado guardado en el csv, el archivo .yaml y el espaciado entre en pixeles que habrá entre los puntos que se objetivo aleatorios a los que se dirigirá el robot.

Codigo patrullar

A partir de aquí se sale del archivo de la interfaz y se pasa al script de patrullaje.py que en primer lugar generarán puntos que cumplan que estén en un área accesible, estén a la distancia especificada de otro y que estén a una distancia de seguridad de la pared.

Código generar_puntos

Luego se seleccionan 5 puntos aleatorios de entre todos los puntos generados.

Código seleccionar_puntos_aleatorios

Estos puntos seleccionados son pixeles aún por lo que hay transfórmalos a coordenadas.

Código convertir_a_coordenadas

Por último, se envían las 5 ubicaciones a la función mover a goal vista anteriormente.

Código patrullar_area

El robot se estará hasta que recorra todos los puntos del área, terminando así la labor de patrulla. A continuación, un ejemplo.

Video del robot patrullando luego de usar el touchpad

Otro método para realizar labores de patrulla es emplear el script de centroides.py que hace el robot recorra todas estancias del área.

El script consiste de tres funciones, siendo la primera la de leer los centroides del csv.

Código leer_centroides

La segunda se encarga de enviar a la función de movimiento los centroides antes leídos para que el robot se dirija a ellos.

Código mover_a_centroides

Y finalmente la función antes vista para usar la librería Move.

Código mover_a_goal

Para acceder a esta funcionalidad solamente hay que pulsar el botón de nombre centroides en la interfa.

Video recorriendo los centroides