

UNIVERSIDAD CARLOS III DE MADRID



ESCUELA POLITÉCNICA SUPERIOR

PROYECTO FIN DE CARRERA
INGENIERO INDUSTRIAL

Título del proyecto
Método de suma de gaussianas para
el problema de scan-matching

Autor:
Piotr Pawel
Jurewicz Slupska

Tutor:
Dr. Luis Moreno Lorente

Leganés, 9 de octubre de 2009

Agradecimientos

Principalmente a mis padres por todo su apoyo estos años de carrera y sobre todo la ayuda dada para dar este último empujón.

A mi novia Paloma por soportarme sobre todo estos últimos meses, y por la ayuda prestada para corregir numerosas faltas de ortografía.

Además, a Luis Moreno por haber dirigido este proyecto de la forma que lo ha hecho y por darme la posibilidad de iniciarme en el apasionante mundo de la investigación.

Por último, quisiera agradecer a todo el mundo que de una forma u otra ha estado implicado en el desarrollo de este trabajo (Jorge, Jaime, Fernando, Alvaro, Nuria y demás), por su paciencia y apoyo.

Otra vez muchas gracias a todos, que sin vosotros esto no sería posible.

Índice general

1. Ideas Básicas	8
1.1. Introducción	8
1.2. ¿Por qué SLAM?	10
1.3. Aplicaciones de SLAM	11
1.4. Necesidad de algoritmos de asociación de datos	12
1.5. Scan-Matching frente a asociación de datos	14
2. Localización autónoma	16
2.1. Mapas de navegación y su aplicación al SLAM	16
2.2. Mapas de ocupación	17
2.3. Mapas de características	19
2.4. Mapas topológicos	21
3. Simultaneous Localization and Mapping SLAM	24
3.1. Filtro de Kalman	24
3.1.1. Espacio de estados	25
3.2. Transformaciones lineales	26
3.3. Filtro de Kalman	27
3.4. Filtro de Kalman Extendido	28
3.5. Algoritmo SLAM Estocástico	29
3.5.1. Modelo del vehículo y del mapa	30
3.5.2. Paso de predicción	31
3.5.3. Paso de actualización	33
3.5.4. Aumento de estado	34
4. Asociación de datos	36
4.1. Asociación de medidas individuales	37
4.1.1. Máxima verosimilitud	37

4.1.2. Compatibilidad Individual	38
4.1.3. Combinación de compatibilidad idividual y máxima verosimilitud	39
4.2. Asociación de paquetes de datos	39
5. Scan-Matching	41
5.1. Definición del problema	42
5.1.1. La estimación de la pose mediante el alineamiento de barridos	42
5.1.2. Criterio para alinear barridos	42
5.2. Técnicas de Scan matching	43
5.2.1. Alineado basado en la correspondencia de puntos	43
5.2.2. Representación probabilística de los datos	48
5.2.3. Representación Bayesiana de la función de verosimilitud para el modelo punto-objetivo (<i>point-target</i>)	51
5.2.4. Correlación de suma de gaussianas	54
6. Resultados experimentales	62
6.1. Algoritmo	62
6.2. Resultados	63
6.2.1. Datos de Bailey	64
6.2.2. Datos del laboratorio Intel	67
6.2.3. Resultados de los datos de Maggie	68
6.3. Resultados de tiempos de computación	78
7. Conclusiones y trabajo futuro	79

Índice de figuras

1.1.	Robot explorador <i>Oportunity</i> de la NASA (2006) en una misión a Marte	11
1.2.	Ambigüedad debida a la incertidumbre en la medida. En la figura, la elipse segmentada indica una pequeña incertidumbre en la medida, mientras que la elipse grande indica una gran incertidumbre. Cuanto mayor sea la incertidumbre, mayor cantidad de características entrarán en la asociación	13
1.3.	Ambigüedad debida a la incertidumbre en la pose. Aquí la elipse rodea la estimación de la pose indicando incertidumbre en la posición del robot. Mientras que las líneas punteadas indican la incertidumbre de hacia dónde está apuntando. Entonces dadas estas dos incertidumbres para la medida Z podría estar asociado con F_1 o con F_2	13
1.4.	Ambigüedad debida a incertidumbre en los puntos de referencia. Aquí la incertidumbre en F_1 y F_2 hace que Z podría estar asociada a cualquiera de ellos	14
2.1.	Mapa de matriz de ocupación. La probabilidad de que una celda esté ocupada está limitada por $(0, 1)$ de forma que 0 indica que no está ocupada y 1 que lo está. Una probabilidad de 0.5 es un espacio no explorado, en la imagen gris claro.	18
2.2.	Mapa de características. El entorno es definido por una serie de características estáticas, en este caso puntos, y estos son utilizados para calcular el movimiento del robot	19

2.3.	Correlación de incertidumbres en SLAM. En (a) el vehiculo observa e inicializa la característica f_1 y la estimación de f_1 es dependiente de la estimación de la pose del vehiculo. En el paso siguiente (b) el vehiculo observa f_1 e inicializa f_2 . La estimación de f_2 depende de la pose del vehiculo en ese momento, que a su vez es dependiente de f_1 . Cuantas más características son observadas, más correlaciones van apareciendo, y el mapa se va haciendo más y más rígido	20
2.4.	Mapa topológico. El entorno es definido usando un gráfico de la estructura de los datos, dónde cada nodo contiene la descripción de un lugar y una descripción del camino para llegar a el. Trasladarse de un punto a otro se hace siempre a través de puntos intermedios, y buscando siempre el camino más corto. De esta forma viajar de A hasta H se haría por la ruta A-D-F-H.	21
3.1.	Algoritmo SLAM genérico	25
3.2.	Vector de estado aumentado. El vector de estado en SLAM está compuesto de la pose del vehículo y de los puntos observados en el entorno referenciados a un sistema de coordenadas global	31
3.3.	Cambio en el vector de la pose. Existen muchos algoritmos para calcular la pose una vez movido el robot, aunque los más utilizados son los relacionados con asociación de datos.	32
5.1.	Datos en bruto de un túnel de una mina. En este entorno es complicado extraer características geométricas. En este tipo de entornos resultan muy útiles técnicas de scan-matching.	41
5.2.	Resultados de puntos más cercanos usando ICP. Se puede ver en la imagen que no se tiene mucha información sobre la rotación.	44
5.3.	Regla para matching-range-point. Para un punto P, el correspondiente P' en el barrido está en el sector y es es punto más cercano a P'	45
5.4.	Puntos más cercanos usando la técnica IMRP. Se puede observar que se obtiene más información sobre el giro que con el algoritmo ICP	46
5.5.	Localización del robot en dos dimensiones. Un único punto de referencia es guardado en el mapa como (x_m, y_m) , y observado por el robot por la medida (x_z, y_z)	52
5.6.	Datos obtenidos del láser sin procesar	55
5.7.	Representación de igual volumen. Los picos representan medidas con poca incertidumbre	56

5.8. Representación de igual altura. Estipula que las medidas muy precisas contienen menos información sobre la forma del entorno, y produce mejor descripción del espacio	57
5.9. algoritmo para cálculo de la verosimilitud	61
5.10. Función de verosimilitud con probabilidad de detección de falsa medida. En la figura (a) se observa un a función de verosimilitud nominal, mientras que la (b) tiene probabilidad de falsa medida	61
6.1. Diagrama de flujo del algoritmo de la suma de gaussianas	63
6.2. Barrido de referencia para los datos de Tim Bailey.	64
6.3. Datos a lo largo del pasillo de Tim Bailey	65
6.4. Error cometido en el giro de Tim Bailey	65
6.5. Vuelta a uno de los puntos iniciales de Tim Bailey	66
6.6. Error cometido en el cierre del bucle de Tim Bailey	66
6.7. Mapa de laboratorio Intel	67
6.8. Mapa de laboratorio Intel obtenido con los datos aplicando el algoritmo SOG. Se puede apreciar el error cometido con respecto al original	69
6.9. Resultado para una vuelta completa con desplazamiento de 5cm	69
6.10. Detalle de cierre de lazo para 5cm	70
6.11. Resultado de una vuelta completa con distanciamiento entre medidas de 10 cm	70
6.12. Detalle de cierre de lazo para 10 cm. Se puede observar que el error cometido es más grande que para 5 cm	71
6.13. Resultado de vuelta completa con distanciamiento entre medidas de 20cm	71
6.14. Detalle de cierre de lazo para 10 cm. Se puede observar que el error cometido es más grande que para 10 cm	72
6.15. Error cometido con un desplazamiento de 30 cm	72
6.16. Vuelta completa con un desplazamiento de 30 cm	73
6.17. Error que se comete con un desplazamiento de 40 cm	73
6.18. Vuelta completa con un desplazamiento de 40 cm	74
6.19. Error cometido con un desplazamiento de 50 cm	74
6.20. Vuelta completa con un desplazamiento de 50 cm. Se puede ver que ya va degenerando demasiado y no se aparece a los resultados de los otros desplazamientos	75
6.21. Error cometido con un desplazamiento de 100 cm	75

6.22. Vuelta completa con un desplazamiento de 100 cm. Se puede ver que los resultados no se parecen en nada con los de los otros desplazamientos. . .	76
6.23. Barrido de referencia para Maggie.	76
6.24. Primer giro de Maggie	77
6.25. Error de Maggie al salir de la habitación	77
6.26. Resultados de computación para el algoritmo SOG	78

Capítulo 1

Ideas Básicas

1.1. Introducción

La robótica es el arte de percibir y manipular a través de dispositivos controlados por ordenador, el mundo real. Ejemplos de robots tendremos por todas partes: robots de exploración de planetas, robots de producción, cirugía asistida, etc. Los robots captan datos del mundo a través de sensores y actúan gracias a actuadores.

Hay que reconocer que la robótica, pese a que se lleva estudiando unos 25 años, esta en sus principios. A día de hoy pensar en robots que nos ayuden en nuestras tareas cotidianas es una fantasía, pero cada vez más próxima. El problema es que no nos movemos en un mundo predecible, hay cosas que cambian rápidamente, y por ello hay que buscar que los robots sean capaces de detectar estos cambios y puedan tomar un cambio de decisión.

Para empezar los sensores que se utilizan en la actualidad tienen limitadas prestaciones, aunque es cierto que cada vez más nos acercamos a sensores más y más precisos, no deja de existir una incertidumbre en la medida que en algunos casos puede llevar al robot a tomar decisiones erróneas. Y también los sensores se pueden estropear, y detectar que un sensor ha fallado no siempre es una tarea fácil.

Los actuadores incluyen motores, que en muchos casos también tienen incertidumbre. La incertidumbre es causa de una mezcla del ruido en el control, fallos mecánicos, etc. Tener altas precisiones implica elevados costes, así la elección dependerá de la precisión requerida para cada aplicación.

Y sobre otras fuentes de incertidumbre se puede comentar pues fallos en el software,

aproximaciones tomadas a la hora de elaborar los algoritmos, cambios en el terreno, etc. El nivel de incertidumbre dependerá de la aplicación. Si es cierto que este nivel será mas bajo en cadenas de montaje y será mayor en entornos domésticos pero no dejara de existir. Pero como la robótica esta yendo a la interacción en espacios abiertos, el mayor de los retos es gestionar bien esta incertidumbre.

Un robot tiene que saber dónde está en un entorno para poder moverse autónomamente e inteligentemente. La auto-localización requiere de la existencia de un mapa, y es lo que ha llevado al desarrollo de algoritmos de SLAM(Simultaneous Localization and Mapping) a lo largo de los últimos 20 años, dónde el robot va construyendo partes del mapa mientras va explorando el entorno. La forma más predominante de SLAM es la estocástica. Este método tiene explícitamente en cuenta el error que se comete en las medidas; estos errores de medida inducen incertidumbre en las estimaciones del entorno, que consiguientemente conllevan a incertidumbres en la posición del robot. Por lo cual se puede decir que el mapa del entorno y la pose están directamente relacionados. Las implementaciones de SLAM estocástico representan estas incertidumbres y correlaciones mediante la distribución gaussiana (PDF), y propagan las incertidumbres usando el filtro de Kalman extendido(Extended Kalman Filter) (**EKF**). Esto es conocido como EKF-SLAM. Un problema del EKF-SLAM es que requiere modelos geométricos del entorno, lo que limita las aproximaciones deseadas para tales entornos[3].

Una alternativa a estos modelos son los procedimientos llamados scan-correlation. Con estos métodos se calcula el máximo alineamiento que hay entre 2 series de datos. Entonces, dadas una serie de observaciones y un mapa de referencia construido también de datos sin tratar, el robot se puede localizar a sí mismo sin la necesidad de convertir los datos a ningún tipo de forma geométrica. Las observaciones son simplemente alineadas con el mapa de datos hasta maximizar la correlación de la medida. Scan-correlation ha sido usado como algoritmo de localización para mapas a priori, siendo el algoritmo ICP o la correlación de matriz de ocupación los métodos más utilizados. Este trabajo consistirá en el estudio y análisis del método de la suma de gaussianas y ver cuales son sus prestaciones.

1.2. ¿Por qué SLAM?

En situaciones dónde el robot se mueve en un entorno con el objetivo de localizarse a si mismo y recopilar información de sus alrededores, por ejemplo para mapeo o seguimiento de objetivos, el tema de SLAM se hace necesario.

Sin embargo, no siempre es óptimo resolver todo el SLAM, dado que se pueden dar situaciones dónde sólo se requiera localización o solo mapeo, en este tipo de situaciones se puede discutir si es o no necesario: por ejemplo cuando se opere en lugares dónde funcionen bien las señales de GPS, como puede ser sistemas aéreos o en campo abierto sobre el suelo, resolver el SLAM no sería necesario para tener delimitada la posición del robot, dado que un sensor nos facilita toda esa información. Pero por desgracia, existen muchos lugares dónde la señal de GPS no es buena, como entornos cerrados o grandes ciudades, por lo cual hay que abordar los problemas desde otro lado. Supongamos por ejemplo un caso en el cual solo se requiere auto-localización en un entorno en el cual no se recibe ninguna señal GPS, si no se va tratar como un problema de SLAM, sólomente se podrá conocer la posición con la ayuda de aparatos de medida o sensores inerciales, tales como acelerómetros, giróscopos, encoders, etc. Incluso aunque se usen combinaciones de estos, hacer una localización on-line resulta complicado, dado que pequeños errores de medida se propagan y hacen que el error acumulado crezca. Este problema sin observadores de estado, van a desembocar en filtros de navegación que diverjan, en otras palabras, la precisión en la estimación de estado empeorará rápido. En algunas de estas situaciones estos problemas se pueden aliviar, o por lo menos retrasar, introduciendo algunas restricciones en partes del filtro de navegación.

Si por el contrario solo se necesita construir un mapa o encontrar un objetivo usando sensores de barrido, construir un mapa sin conocer la posición no será posible. En situaciones en las que se tiene buena información *a priori* referente a la posición del robot, entonces la solución de un mapa es abordable. Sin embargo, tan pronto como el robot se empiece a mover la incertidumbre en la localización empezará a crecer sin límites. Luego parece lógico que para este tipo de problemáticas se emplee métodos de SLAM.

Otra motivación para usar SLAM es que se puede implementar sin excesivo coste, al contrario que si se quisiera hacer por separado dado que habría que montar sistemas muy complejos, que a su vez no aseguran la correcta coherencia de los datos. Si se



Figura 1.1: Robot explorador *Opportunity* de la NASA (2006) en una misión a Marte

hace de una manera adecuada se puede implementar una solución SLAM con poca instrumentación. Además, resolver un SLAM puede ser hecho de manera autónoma sin la necesidad de un usuario, luego puede decirse que no hace falta un usuario que use un instrumental[2].

1.3. Aplicaciones de SLAM

Existen gran cantidad de aplicaciones dónde la implantación de SLAM puede ser de gran ayuda tanto en el ámbito humano como en el robótico. Es por ejemplo, un requisito para el correcto funcionamiento de robots móviles, dado que la mayoría de estos tiene implementada alguna forma de SLAM. Algunas aplicaciones interesantes pueden ser vehículos que operen en las profundidades marinas, en la exploración de minas, exploración de planetas, etc. Para la mayoría de estas tareas el SLAM es usado solamente para temas de navegación, sin embargo algunos usan el SLAM para recopilar información del entorno para trabajos futuros.

También se puede usar el SLAM para mejorar las capacidades de los humanos en temas de localización y mapeo, y en este aspecto tiene un interés muy importante. Se supone que sistemas bien implementados combinados con aparatos de medida precisos, puedan dar lugar a robots que bien trabajen bajo la supervisión humana, bien trabajen de manera autónoma. Este tipo de sistemas pueden ser usados en lugares que necesiten realizar tareas complejas para ser útiles, o que requieran una instrumentación muy cara. Un

ejemplo de esto está en los aparatos que usa el ejército para localizar objetivos, que son aparatos caros y voluminosos, y a demás requieran el uso de personal.

Por esto las aplicaciones del SLAM en el mundo militar son de gran interés, para hacer mapas de zonas en las que en un futuro se vayan a hacer operaciones, explorar zonas por las cuales vayan a pasar tropas, explorar objetivos desconocidos, etc. En general en todo tipo de operaciones en las cuales el uso de GPS se hace imposible.

En aplicaciones civiles tiene interés también en temas de búsqueda de desaparecidos, en montañas, cuevas, etc[2].

1.4. Necesidad de algoritmos de asociación de datos

Hasta no hace mucho tiempo los trabajos sobre SLAM ignoraban el tema de la asociación de datos suponiendo que era siempre conocida *a priori* para todas las medidas. En cualquier situación en la cual un mapa contenga más de un objeto, existirá siempre una cierta ambigüedad en la asociación de datos. De hecho, aunque sólo exista un dato se puede dar también ambigüedad si existe la posibilidad de que existan medidas falsas. En la mayoría de las aplicaciones SLAM el número de dimensiones necesarias para definir el entorno es elevada, dado que el mapa tiene que contener muchas características para que el entorno esté correctamente definido.

Aparte de la ambigüedad creada por la asignación de distintas posibilidades para las medidas individuales, o la asignación de múltiples medidas, existe la ambigüedad creada por la incertidumbre de la medida, de la pose, y de los puntos de referencia. Para describir, como cada tipo de ambigüedad afecta a la asociación de datos, a continuación se van a mostrar unas figuras a modo de introducción. En ellas se usará la siguiente notación: F_i es la estimación correcta de la posición de un punto i , Z es usado para definir la medida, y S_t es la estimación actual de la pose.

Cuanto mayor sea la incertidumbre en la medida, mayor será la ambigüedad en la asociación de datos dado que mayores incertidumbres en la medida significan que se tiene que tener en cuenta mayor número de asociaciones. Esto ocurre porque más asociaciones de datos tendrán verosimilitudes que no puedan ser ignoradas, que es un método utilizado por muchos algoritmos de asociación de datos.

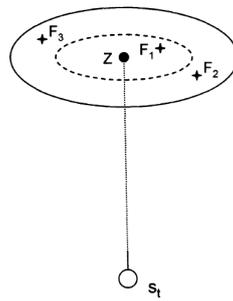


Figura 1.2: Ambigüedad debida a la incertidumbre en la medida. En la figura, la elipse segmentada indica una pequeña incertidumbre en la medida, mientras que la elipse grande indica una gran incertidumbre. Cuanto mayor sea la incertidumbre, mayor cantidad de características entrarán en la asociación

La incertidumbre de la pose tiene un efecto parecido, pero en este caso no está claro que objeto o objetos están siendo observados, dado que no se puede saber con total certeza dónde está o a qué está enfocando el robot, como se puede ver en la Figura 1.2. Finalmente,

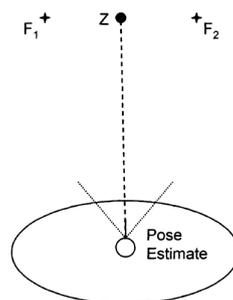


Figura 1.3: Ambigüedad debida a la incertidumbre en la pose. Aquí la elipse rodea la estimación de la pose indicando incertidumbre en la posición del robot. Mientras que las líneas punteadas indican la incertidumbre de hacia dónde está apuntando. Entonces dadas estas dos incertidumbres para la medida Z podría estar asociado con F_1 o con F_2

la incertidumbre en los puntos característicos o de referencia, conlleva que distintos puntos puedan ser asociados a una sola medida. En la siguiente figura se puede observar esta situación. En la práctica todos estos efectos aparecen a la vez, superponiéndose unos a otros. Además, a menudo se da el caso en el cual hay que considerar si una medida es un objeto nunca antes visto o si es una medida falsa, y si es este último caso habría que eliminarla. La combinación de todas estas cosas hace la asociación de datos una tarea muy compleja, y probablemente una de las más importantes del algoritmo SLAM. En capítulos posteriores se tratará más en detalle estos problemas.

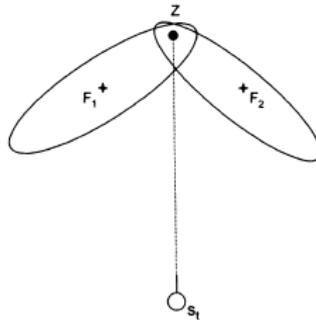


Figura 1.4: Ambigüedad debida a incertidumbre en los puntos de referencia. Aquí la incertidumbre en F_1 y F_2 hace que Z podría estar asociada a cualquiera de ellos

1.5. Scan-Matching frente a asociación de datos

La asociación de datos por medio de las características es una solución viable en entornos donde existan puntos de referencia estáticos que sean aptos para ser clasificados según primitivas geométricas. Sin embargo, en ciertos entornos en los que puede ser difícil definir modelos geométricos, se puede hacer que una asociación de datos sea precisa mediante la correlación directa de los datos obtenidos de las observaciones. La correlación directa también puede ser una alternativa útil en entornos aptos para la asociación de características, dado que incorpora toda la información disponible a través de los sensores y evita clasificaciones de modelos no del todo correctas.

La correlación de datos en bruto (en inglés "raw data"), es también denominada scan-matching, y es el proceso de alineado los puntos observados en un scan o barrido (2D o 3D) respecto a un scan de referencia. De esta forma, scan-matching puede ser definida como función de la pose relativa entre dos paquetes de datos.

Para problemas de localización en robots móviles, scan-matching es un mecanismo muy útil para facilitar localización basada en "dead reckoning" de sensores desde un mapa a priori, y para generación de mapas offline. A día de hoy implementar SLAM estocástico en tiempo real resultaría intratable dado el elevado coste computacional, pero recientemente han salido ciertas investigaciones basadas en EM-SLAM y SLAM híbrido topológico métrico [11] que van en esas direcciones.

Existen gran variedad de técnicas que resuelven este problema, como se verá en capítulos siguientes. Sin embargo, la mayoría son basadas en funciones heurísticas costosas y que

carecen de cualquier derivación probabilística. En este trabajo se estudiará un método que se deriva de un modelo estocástico del sensor, para resolver este problema, de forma que el resultado sea una estimación correcta de la incertidumbre de la pose.

Capítulo 2

Localización autónoma

El problema de la localización autónoma ha sido bastante tratado a lo largo de las dos últimas décadas, como resultado de una gran variedad de paradigmas que fueron surgiendo para determinar la posición y la orientación de un robot móvil en relación con otros objetos del entorno. Algo que tienen en común todos los métodos de localización es un mapa. Este mapa es una representación del entorno que sirve de referencia para la información extraída de los sensores del robot. En este capítulo se tratarán los distintos tipos de mapas que se pueden utilizar.

2.1. Mapas de navegación y su aplicación al SLAM

En este apartado se tratará los tres tipos de mapas usados con más frecuencia en sistemas de localización: matrices de ocupación (grid maps), mapas de características (feature maps) y mapas topológicos. Para cada tipo de mapa se hará una breve descripción funcional y si sirven para la localización con un mapa a priori. Algunos criterios son los mostrados a continuación:

- Representación de la incertidumbre: Los sensores del robot no pueden medir con total exactitud el entorno, luego existirá un cierto grado de incertidumbre en la representación. Al mismo tiempo, la localización del vehículo se deriva de este mapa, luego su pose estimada también tendrá incertidumbre. En esencia, un modelo de incertidumbre tiene que reflejar con fidelidad el error entre el estado actual y el estimado del sistema.
- Convergencia monótona. El primer objetivo de la medida de la incertidumbre es asegurar la convergencia del mapa. Un mapa es convergente cuando si la geometría

espacial del entorno se aproxima a la geometría real cada vez que nuevas observaciones son incorporadas. Esto es lo mismo que decir reducción monótona de la incertidumbre del mapa. Sin esta medida de la incertidumbre, un objeto estático con una localización estimada (x_1, y_1) pueden ser desplazadas con las actualizaciones del mapa a una posición distinta (x_2, y_2)

- Asociación de datos. La representación del mapa tiene que permitir correspondencia fidedigna entre la información obtenida de los sensores del robot y la información almacenada en el mapa. Primero la búsqueda de observación a mapa tiene que ser lo suficientemente eficiente para operación en tiempo real y , segundo, la asociación tiene que ser robusta para visiones parciales y grandes espacios de búsqueda. Visiones parciales se pueden dar cuando una observación consta de una parte ya mapeada, una región sin explorar y objetos dinámicos.
- Detección de ciclo (Cycle detection). Cuando un robot explora un entorno mediante la trazada de un gran bucle, entonces identificar que vuelve a una región de mapa ya explorada es llama detección de ciclo. La detección de ciclo incorpora condiciones especiales a los dos criterios anteriores. El primero la asociación de datos, que es distinto de la asociación local dada la mayor incertidumbre de la pose del vehículo y por tanto espacio de búsqueda involucrado. La eficiencia de búsqueda es un aspecto importante pero, aun mas lo es la robustez en la decisión de si una asociación es correcta o no. Una vez encontrada una asociación correcta, el error acumulado en el mapa tiene que ser compensado correctamente cada vez que se actualice el mapa.
- Tiempo de computación y almacenado. El mapa tiene que almacenar suficiente información para permitir asociación de datos y convergencia. Este almacenaje, y el tiempo de computación tienen que ser proporcionales al área tratada.

2.2. Mapas de ocupación

Los mapas de ocupación son matrices en las cuales porciones del espacio están representadas. Cada celda representa una pequeña porción del espacio e indica la probabilidad de que esa región esté ocupada o no con valores entre $(0,1)$. La localización se hace comparando los datos de cada observación con el mapa usando técnicas de correlación cruzada. Un mapa de este estilo seria el que se puede ver en la Figura 2.1.

Al tratarse de un mapa a priori, las matrices de ocupación son bastante efectivas como

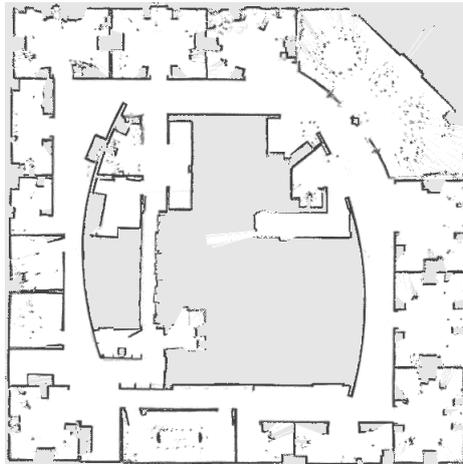


Figura 2.1: Mapa de matriz de ocupación. La probabilidad de que una celda esté ocupada está limitada por $(0, 1)$ de forma que 0 indica que no está ocupada y 1 que lo está. Un probabilidad de 0.5 es un espacio no explorado, en la imagen gris claro.

métodos de localización en entornos pequeños. Esta técnica también concede al usuario información explícita tanto del espacio libre como del ocupado, lo cual es muy útil para la planificación de trayectorias.

Un problema con los mapas de ocupación es la dificultad para asociar datos. La búsqueda por correlación cruzada, en la región en la que se encuentre el robot, requiere mucho coste computacional si el espacio de búsqueda es grande, aunque recientemente se hayan presentado métodos de rápidos de búsqueda[?]. Además si la correlación cruzada resulta multi-modal en el espacio de búsqueda, la búsqueda de máxima probabilidad de correlación fallará. Este problema tiene que ser tratado utilizando el método de Monte Carlo, al cual se hace referencia en la bibliografía.

Pero la dificultad más importante en este tipo de mapas, sobre todo en lo que conciernen a grandes espacios, es la relación entre la resolución de la malla y el tiempo computacional. Lo ideal sería hacer la malla lo más pequeña posible, pero ello conllevaría un aumento muy grande del tiempo de computación. En cambio si se eligen celdas demasiado grandes, se perderá información. También, tareas como la planificación de trazadas pueden elevar su tiempo de computación si el mallado es muy pequeño. Existen métodos de tamaño variable de malla que permite centrarse en las zonas de interés del mapa, pero tienen sus propias dificultades y son difíciles de implementar.

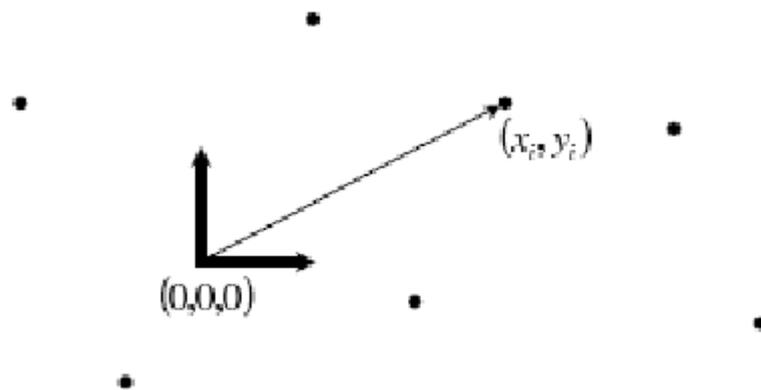


Figura 2.2: Mapa de características. El entorno es definido por una serie de características estáticas, en este caso puntos, y estos son utilizados para calcular el movimiento del robot

La técnica de Occupancy grid SLAM intercala los pasos de localización y actualización del mapa primero registrando el mapa a corto plazo y luego fundiéndolo con el mapa global(localización), y luego actualizar la ocupación percibida del mapa global(mapeo). Este método funciona muy bien para entornos dinámicos pequeños en periodos breves de tiempo. Sin embargo, este método no posee un modelo exacto de incertidumbre por lo que en periodos mas largos tiende a diverger[1].

2.3. Mapas de caraterísticas

Los mapas de características representan el entorno mediante la localización global de ciertas características paramétricas tales como puntos o líneas como se muestra en la figura 2.2. La localización se lleva a cabo extrayendo primero características de los datos observados y asociándolos a características en el mapa. Las diferencias entre las características predichas y los lugares observados son usados para calcular la pose del vehículo. En este sentido, la localización es similar a cazar, pero en este caso la presa esta inmóvil y es el cazador quien se mueve. Los puntos característicos en un mapa de características a priori se asume que son perfectamente conocidos, y cada uno de ellos es definido por sus correspondientes parámetros. Por ejemplo, un punto característico para un entorno cilindrico, como un tubo, podría ser definido como $f=(t,r,x,y)$, dónde t es el tipo de entorno(cilíndrico), r es el radio y x e y definen la posición del centro del vehículo. Solo es la posición del vehículo es directamente útil para la localización, pero la otra información

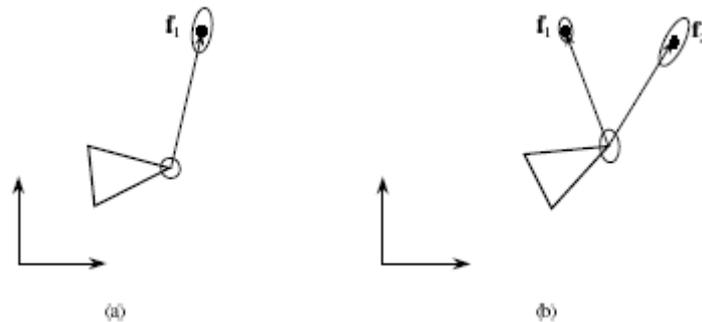


Figura 2.3: Correlación de incertidumbres en SLAM. En (a) el vehículo observa e inicializa la característica f_1 y la estimación de f_1 es dependiente de la estimación de la pose del vehículo. En el paso siguiente (b) el vehículo observa f_1 e inicializa f_2 . La estimación de f_2 depende de la pose del vehículo en ese momento, que a su vez es dependiente de f_1 . Cuantas más características son observadas, más correlaciones van apareciendo, y el mapa se va haciendo más y más rígido

puede servir para la asociación de datos. Al contrario que las matrices de ocupación, donde se lleva a cabo una descripción densa del entorno, los mapas de características son unos mapas bastante eficientes. En concreto, el espacio libre no es representado y por ello no añade ningún coste de computación adicional proceso de localización. Por este motivo, este tipo de mapas no favorece la planificación de rutas y por tanto tienen que ser abordadas como problemas diferentes.

La asociación de datos es el principal problema de este tipo de mapas. La correcta estimación de la pose depende de encontrar una correcta correspondencia entre el punto característico observado y el asociado al mapa. Una asociación errónea daría como resultado una reducción de la incertidumbre sobre la pose del vehículo, pero el error estimado aumentaría.

Otro problema de este tipo de mapas es que son solamente recomendables para entornos donde los objetos pueden ser razonablemente fácil descritos por un modelo geométrico.

Las técnicas featureSLAM hacen 2 tareas simultáneas, añadir las características observadas al mapa, usando la pose del vehículo como referencia, a la vez que usa el mapa acumulado para estimar la posición del vehículo. La incertidumbre del sensor para las

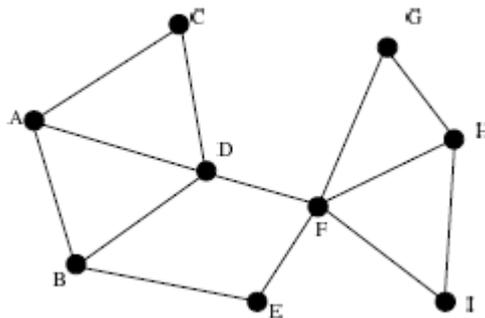


Figura 2.4: Mapa topológico. El entorno es definido usando un gráfico de la estructura de los datos, donde cada nodo contiene la descripción de un lugar y una descripción del camino para llegar a él. Trasladarse de un punto a otro se hace siempre a través de puntos intermedios, y buscando siempre el camino más corto. De esta forma viajar de A hasta H se haría por la ruta A-D-F-H.

medidas, por tanto, resulta también inciertas las estimaciones del vehículo y de los puntos característicos, y estas dos incertidumbres están correladas[1].

En resumen, este tipo de mapas son viables para mapas de gran tamaño, donde haya puntos característicos observables, los tiempos de computación son aceptables.

2.4. Mapas topológicos

Los mapas topológicos son los que tienen mayor carga conceptual, de todos los métodos presentados hasta ahora. Las matrices de ocupación y los mapas de características son ambos mapas métricos donde la localización es definida por una serie de coordenadas cartesianas. Los mapas topológicos, sin embargo, no utilizan medidas métricas sino que representan el entorno como lugares conectados por rutas. Normalmente son descritas de forma gráfica, donde los nodos definen lugares concretos del entorno y los arcos o enlaces definen información sobre la forma de viajar entre ellos. De esta forma viajar de un nodo a otro no conectado a este tiene que hacerse a través de nodos intermedios. El concepto se basa en que los lugares diferenciados son localmente diferenciables del área de alrededor y la información de procedimiento es suficiente para permitir al robot viajar de un punto a otro.

Reconocer los lugares es una forma de asociar datos donde la correspondencia mapa-

observación es basada en la apariencia similar entre los dos juegos de datos- en este caso entre la información observada y la descripción gráfica de nodos. La asociación basada en apariencias, que depende de tener localmente datos únicos, que puede ser considerada en contraste con mapas métricos de proximidad tales como los usados en los mapas de características, que permiten la presencia de puntos de referencia idénticos. Para que el reconocimiento de lugares funcione correctamente, la descripción de cada uno de los nodos tiene que ser distinta, de esta forma el robot compara los datos observados con la plantilla de nodos a la vez que se mueve entre ellos y determinar su localización una vez que halla encontrado un dato que cuadre.

Los mapas topológicos, como referencia a priori, son atractivos dada su eficiencia y que su representación es bastante compacta, y fácil de usar para tareas como la planificación de rutas. Al no tratarse de representaciones métricas la incertidumbre de la pose es un parámetro irrelevante, en cambio se usan medidas cualitativas como sigue la ruta desde A hasta B.

La principal desventaja de esta técnica es que no permite una navegación aceptable entre lugares, y por lo tanto una correcta detección de lugares, sin la ayuda de algún sistema métrico de localización. Moverse de un punto a otro usando solo trayectorias topológicas, como por ejemplo seguir un muro, a veces es suficiente para entornos estáticos y estructurados, pero en entornos dinámicos suelen fallar con gran facilidad. Si un lugar no es reconocido (falso negativo) o es reconocido erróneamente (falso positivo), entonces la secuencia topológica se pierde y el robot se desorienta.

El SLAM topológico opera haciendo la exploración del entorno siguiendo una serie de criterios a la hora de determinar su trayectoria, y guardando las descripciones de los lugares. Cada vez que un lugar nuevo es encontrado, es conectado al lugar anterior siguiendo las especificaciones hasta llegar a él. De esta manera, el mapa es construido como secuencia lineal de lugares, que continúan hasta que un lugar encontrado cuadre con uno anteriormente almacenado. Si este lugar puede ser catalogado como uno anterior entonces se cierra un ciclo o bucle.

Uno de los problemas más importantes para este tipo de técnicas para SLAM es la detección de ciclo. Al no hacerse la localización de punto de forma métrica, se eliminan las dificultades de la incertidumbre en la representación y las no-linealidades, pero por

contra, la parte más crítica es la asociación de datos. En caso de la detección de un ciclo, donde un lugar se asemeja a uno anteriormente almacenado, la asociación de datos pasa a ser ambigua, y el lugar observado puede ser uno de los almacenados anteriormente o uno nuevo que se asemeje. Discernir entre estos dos casos no siempre es fácil, pero existe un método llamado rehearsal[?] que permite hacerlo con cierta facilidad.

Resumiendo, salvo que el entorno posea una única secuencia de exploración, la detección de ciclos será siempre ambigua. Esta es la principal desventaja de este método frente a los estudiados anteriormente. Una posible solución sería introducir algún tipo de información métrica.

Capítulo 3

Simultaneous Localization and Mapping SLAM

El problema de SLAM empieza a ser tratado en profundidad cuando los problemas de mapeo y navegación empezaron a ser muy complicados, y en ciertos casos imposibles de resolver, si se hacían por separado. Los mejores métodos de SLAM fueron los que utilizaban métodos estadísticos para tener en cuenta las incertidumbres de los estados del sistema, así como la incertidumbre en la medida. Este tipo de métodos fueron introducidos por primera vez por Smith y Cheeseman..

El SLAM es un método muy efectivo en entornos donde no hay buena cobertura de GPS tales como aulas, minas, etc. En estas situaciones, tanto la incertidumbre de la posición de los objetos del entorno como la incertidumbre de la pose del robot están fuertemente relacionados. Esto es debido a que en estas situaciones el robot tiene que relacionar las medidas que toma con sus sensores de barrido con su propia pose. Esto lleva a resolver el problema de manera conjunta, y llamarlo SLAM o como se puede encontrar en algunas referencias como *Concurrent Mapping and Localization*. El algoritmo SLAM seguiría de forma muy resumida el diagrama de flujo de la figura 6.1

3.1. Filtro de Kalman

En este capítulo se hará una breve introducción a la representación en el espacio de estados, y se hará una breve descripción del filtro de Kalman lineal y del extendido. Para saber de dónde se deducen las fórmulas, consultar bibliografía[12].

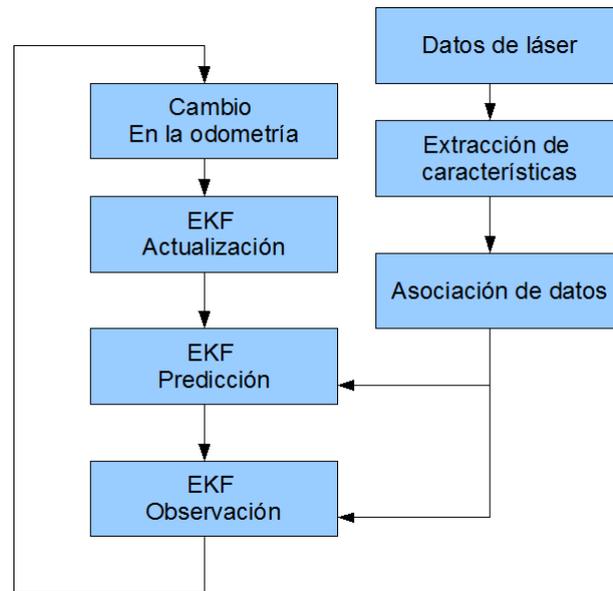


Figura 3.1: Algoritmo SLAM genérico

3.1.1. Espacio de estados

En la representación en el espacio de estados de un sistema, este queda representado mediante una serie de parámetros escalares, llamados variables de estados, de forma que el sistema es definido por un vector de estado de la forma $X = [X_1, \dots, X_n]^T$. Normalmente no se puede conocer con total exactitud el valor del vector de estado, luego existirá una cierta incertidumbre para cada variable de estado. Si esta incertidumbre tiene una distribución normal, entonces cada variable puede ser definida en términos de su media y de su desviación típica σ . Generalmente, la magnitud de la incertidumbre viene dada por la varianza σ^2 , porque de esta forma la incertidumbre de variables independientes se puede añadir. Por ejemplo, si x_1 y x_2 son independientes con unas desviaciones típicas σ_1 y σ_2 , y sea $x_3 = x_1 + x_2$ y $\sigma_3^2 = \sigma_1^2 + \sigma_2^2$. Entonces el sistema x puede ser representado por el vector media y por la matriz de covarianza.

$$\hat{x} = \begin{bmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_n \end{bmatrix} = E[x]$$

$$P_x = \begin{bmatrix} \sigma_{11}^2 & \cdots & \sigma_{1n}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{1n}^2 & \cdots & \sigma_{nn}^2 \end{bmatrix} = E[(x - \hat{x})(x - \hat{x})^T]$$

Donde $E[x]$ es la media definida de la forma $E[x] = \int_{-\infty}^{\infty} xp(x) dx$ la función $p(x)$, representa la densidad de x . Para más información sobre funciones de densidad consultar la

bibliografía [1].

Los términos de la diagonal de P_x son llamados correlaciones cruzadas, y representan la dependencia de incertidumbres entre unas variables y otras.

3.2. Transformaciones lineales

Si x es transformado en y por medio de una serie de sistemas de ecuaciones lineales F , entonces la matriz de covarianza es transformada de la siguiente forma

$$\begin{aligned}\hat{y} &= F\hat{x} \\ P_y &= FP_xF^T\end{aligned}$$

La suma de dos vectores de estado independientes se haría de forma similar

$$\begin{aligned}\hat{x}_3 &= A\hat{x}_1 + B\hat{x}_2 \\ P_3 &= AP_1A^T + BP_2B^T\end{aligned}$$

Para transformaciones no lineales, sin embargo, existe un problema, y no es que no sea posible transformar la matriz de covarianzas de un estado a otro. La solución más típica para este problema es aproximar las transformaciones de la matriz de covarianzas linealizando las funciones de transformación. Dadas unas funciones no-lineales f , la matriz jacobiana linealizada ∇f_x es definida como la derivada de f con respecto de x en el punto \hat{x} .

$$\begin{aligned}y = f(x) &= \begin{bmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_m(x_1, \dots, x_n) \end{bmatrix} \\ \nabla f_x = \frac{\partial f}{\partial x} \Big|_{\hat{x}} &= \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}\end{aligned}$$

Asumiendo que f es razonablemente lineal en una región pequeña, la función linealizada de la matriz se aproxima a f para valores cercanos de x para \hat{x} . De esta forma, la transformación de la covarianza puede ser calculada de la siguiente forma

$$P_y = \nabla f_x P_x \nabla f_x^T \quad (3.1)$$

Existe un problema con estas aproximaciones, y es que las transformaciones introducen un "bias." estimación de la covarianza y esto puede llevar a incurrir en un error

importante si la función es muy no-lineal.

Un ejemplo de transformación podría ser pasar de coordenadas polares a coordenadas cartesianas. Unas medidas polares p , como las que devuelve un sensor láser, a menudo es útil tenerlas en coordenadas cartesianas

$$\hat{p} = \begin{bmatrix} \hat{r} \\ \hat{\phi} \end{bmatrix} \xrightarrow{f} x = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} \quad (3.2)$$

$$P_p = \begin{bmatrix} \sigma_{rr}^2 & \sigma_{r\phi}^2 \\ \sigma_{\phi r}^2 & \sigma_{\phi\phi}^2 \end{bmatrix} \xrightarrow{f} P_x \quad (3.3)$$

donde la transformación no lineal f viene dada por

$$x = f(p) = \begin{bmatrix} r \cos \phi \\ r \sin \phi \end{bmatrix} \quad (3.4)$$

La estimación transformada \hat{x} puede ser calculada directamente de la función no lineal pero la covarianza tiene que ser calculada de la jacobiana

$$\nabla f_p = \left. \frac{\partial f}{\partial p} \right|_{\hat{p}} = \begin{bmatrix} \cos \hat{\phi} & \hat{r} \sin \hat{\phi} \\ \sin \hat{\phi} & \hat{r} \hat{\phi} \end{bmatrix}$$

$$P_x = \nabla f_p P_p \nabla f_p^T$$

3.3. Filtro de Kalman

El filtro de Kalman es un algoritmo que fusiona información en el espacio de estados. Esto es aplicable a sistemas donde tanto las transiciones de estado y las observaciones de estado son lineales y que las incertidumbres asociadas son gaussianas. Esencialmente, funciona como un ciclo de 2 pasos, un paso de predicción y otro de actualización.

Dado un sistema definido por un estado x_k (en un instante k) y modelado por el proceso

$$x_{k+1} = Ax_k + Bu_k + q_k \quad (3.5)$$

donde u_k representa algún vector de estado independiente (normalmente una entrada de control) y q_k es ruido blanco gaussiano, el paso de la predicción se rige por las siguientes ecuaciones

$$x_{k+1}^- = Ax_k + Bu_k$$

$$P_{k+1}^- = AP_k A^T + BU_k B^T + Q_k$$

El signo - indica que es una predicción de estado a priori. Las matrices P_k, U_k y Q_k son las matrices de las covarianzas de x_k, u_k y q_k respectivamente. Si, en un instante $k+1$ se hace una observación de estado independiente

$$z_{k+1} = Hx_{k+1} + r_{k+1} \quad (3.6)$$

en el que r_{k+1} representa ruido blanco gaussiano con una covarianza R_{k+1} , entonces se puede mejorar la estimación de estado. Esto es, la estimación actualizada de x_{k+1} será la suma ponderada del estado predicho y del estado observado. El filtro de Kalman determina la ponderación óptima cómo:

$$\begin{aligned} \nu_{k+1} &= z_{k+1} - Hx_{k+1}^- \\ S_{k+1} &= HP_{k+1}^- + R_{k+1} \\ W_{k+1} &= P_{k+1}^- = P_{k+1}^- - W_{k+1}S_{k+1}W_{k+1}^T \end{aligned}$$

$$\begin{aligned} x_{k+1}^+ &= x_{k+1}^- + W_{k+1}\nu_{k+1} \\ P_{k+1}^+ &= P_{k+1}^- - W_{k+1}S_{k+1}W_{k+1}^T \end{aligned}$$

Cabe destacar que el filtro de Kalman estima la covarianza P_{k+1} independientemente del valor de la media de estado x_{k+1} . De hecho, para sistemas lineales, es posible calcular la covarianza de estado offline [12].

3.4. Filtro de Kalman Extendido

En la mayoría de sistemas reales, no son gobernados por ecuaciones lineales, el filtro básico de Kalman no es suficiente para esas tareas. Para tratar sistemas de ecuaciones no lineales se tiene que usar el filtro de Kalman extendido. Por ejemplo, una transición de estado modelada por una función no lineal f y una observación del estado modelada por una función no lineal h como se muestra a continuación

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + q_k \\ z_{k+1} &= h(x_{k+1}) + r_{k+1} \end{aligned}$$

La clave del EKF (Extended Kalman Filter) está en linealizar las funciones f y h sobre el punto de la media del estado. De esta forma, y al contrario que el filtro de Kalman, la estimación de la covarianza de estado es dependiente de la estimación de la media. El

paso de la predicción pasa a ser

$$x_{k+1}^- = f(\hat{x}_k, \hat{u}_k)$$

$$P_{k+1}^- = \nabla F_{x_k} P_k \nabla f_{x_k}^T + \nabla f_{u_k} U_k \nabla f_{u_k}^T + Q_k$$

donde la jacobiana ∇f_{x_k} es la derivada parcial de f con respecto a x en el punto (\hat{x}_k, \hat{u}_k) y la jacobiana ∇f_{u_k} es determinada de manera similar para u

$$\nabla f_{x_k} = \frac{\partial f}{\partial x_k}$$

$$\nabla f_{u_k} = \frac{\partial f}{\partial u_k}$$

Para calcular la ganancia de Kalman se haría de manera similar pero linealizando la función de observación h

$$\nu_{k+1} = z_{k+1} - H x_{k+1}^-$$

$$S_{k+1} = \nabla h_{k+1} P_{k+1}^- \nabla h_{k+1}^T + R_k$$

$$W_{k+1} = P_{k+1}^- \nabla x_{k+1}^T S_{k+1}^{-1}$$

Dónde la jacobiana ∇h_{k+1} viene dada por

$$\nabla h_{k+1} = \frac{\partial h}{\partial x_{k+1}} \quad (3.7)$$

La actualización de estado se hace de la misma manera que en el filtro de Kalman simple. El EKF se basa en asumir que las funciones h y f son quasi-lineales de forma que las transformaciones linealizadas son lo suficientemente aproximadas como para dar transformaciones de covarianza correctas. Sin embargo, para funciones fuertemente no lineales la linealización puede generar graves errores en la estimación de la incertidumbre, por lo que habría que utilizar otras técnicas, como pueden ser los filtros de partículas[12].

3.5. Algoritmo SLAM Estocástico

En este capítulo se presentarán las ecuaciones básicas para la implementación de SLAM con mapas de características basadas en el EKF (Extended Kalman Filter, que se explicó en el capítulo anterior). Es importante entender bien el filtro de Kalman para seguir los pasos del SLAM estocástico. Este SLAM se desarrolla almacenando la pose del vehículo y los puntos de referencia del mapa en un solo vector de estado, y estimando los parámetros de estado vía procesos recursivos de observación y predicción. En el proceso

de predicción se trabaja con el movimiento del robot, lo cual incrementa la incertidumbre de la estimación de la pose del vehículo. El proceso de observación, o actualización, se hace re observando las características almacenadas, lo cual mejora la estimación de estado global. Cuando una característica es detectada por primera vez, es añadida al vector de estado por medio de un proceso llamado de inicialización.

3.5.1. Modelo del vehículo y del mapa

El estado del vehículo es representado a través de su posición relativa respecto a un eje de coordenadas cartesianas (como se puede ver en la figura 3.2) con una media y una covarianza definida como

$$\hat{x}_v = \begin{pmatrix} \hat{x}_v & \hat{y}_v & \hat{\phi}_v \end{pmatrix}^T$$

$$P_v = \begin{pmatrix} \sigma_{x_v x_v}^2 & \sigma_{x_v y_v}^2 & \sigma_{x_v \phi_v}^2 \\ \sigma_{x_v y_v}^2 & \sigma_{y_v y_v}^2 & \sigma_{y_v \phi_v}^2 \\ \sigma_{x_v \phi_v}^2 & \sigma_{y_v \phi_v}^2 & \sigma_{\phi_v \phi_v}^2 \end{pmatrix}$$

Las ubicaciones 2D de puntos característicos observados por el vehículo desde un mapa en el mismo sistema de coordenadas. La matriz de covarianzas P_m de este mapa incluye correlación cruzada que incluye información entre las características que captura la dependencia de la ubicación de cada punto característico sobre el conocimiento de otros puntos del mapa. Mientras estos puntos sean estáticos, estas correlaciones van a ir incrementándose por lo cual el mapa se irá haciendo cada vez más y más rígido.

$$\hat{x}_m = \begin{pmatrix} \hat{x}_1 & \hat{y}_1 & \cdots & \hat{x}_n & \hat{y}_n \end{pmatrix}$$

$$P_m = \begin{pmatrix} \sigma_{x_1 x_1}^2 & \sigma_{x_1 y_1}^2 & \cdots & \sigma_{x_1 x_n}^2 & \sigma_{x_1 y_n}^2 \\ \sigma_{x_1 y_1}^2 & \sigma_{y_1 y_1}^2 & \cdots & \sigma_{y_1 x_n}^2 & \sigma_{y_1 y_n}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_{x_1 x_n}^2 & \sigma_{y_1 x_n}^2 & \cdots & \sigma_{x_n x_n}^2 & \sigma_{x_n y_n}^2 \\ \sigma_{x_1 y_n}^2 & \sigma_{y_1 y_n}^2 & \cdots & \sigma_{x_n y_n}^2 & \sigma_{y_n y_n}^2 \end{pmatrix}$$

El mapa SLAM es definido por el vector aumentado de estados, que es la concatenación del estado del vehículo y del estado de los puntos del mapa. Para tener un SLAM consistente es necesario mantener correlación entre el vehículo y el mapa.

$$\hat{x}_a = \begin{pmatrix} \hat{x}_v \\ \hat{x}_m \end{pmatrix}$$

$$P_a = \begin{pmatrix} P_v & P_{vm} \\ P_{vm}^T & P_m \end{pmatrix}$$

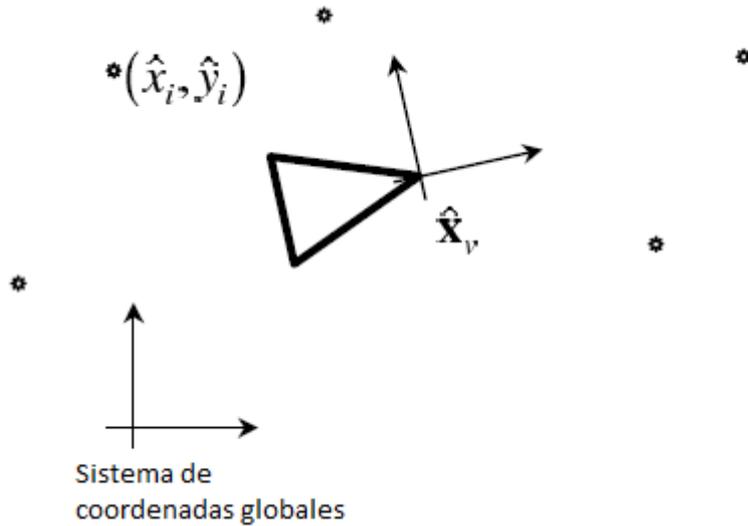


Figura 3.2: Vector de estado aumentado. El vector de estado en SLAM está compuesto de la pose del vehículo y de los puntos observados en el entorno referenciados a un sistema de coordenadas global

Para un estado inicial la estimación de estado suele ser $\hat{x}_a = \hat{x}_v = 0$ y $P_a = P_v = 0$, lo que quiere decir que todavía no se ha observado nada.

3.5.2. Paso de predicción

El modelo del proceso SLAM especifica que el vehículo se mueve con respecto a su pose anterior con respecto una estimación de movimiento (motion), y las características del mapa se mantienen estacionarias. El efecto de este modelo en la estimación de estado es un cambio en X_v , perteneciente al vector de estado, y en P_v y P_{vm} de la matriz de covarianzas, mientras X_m y P_m se mantienen constantes.

Una estimación del cambio de la posición del vehículo \hat{x}_δ con una covarianza P_d (como la figura 3.3) que es obtenida normalmente usando un encoder para obtener la odometría y el modelo cinemático del vehículo. De este modo el vector de estado aumentado predicho será

$$\hat{x}_a^- = f(\hat{x}_a, \hat{x}_\delta) = \begin{pmatrix} g(\hat{x}_v, \hat{x}_\delta) \\ \hat{x}_m \end{pmatrix} = \begin{pmatrix} \hat{x}_v + \hat{x}_\delta \cos \hat{\phi}_v - \hat{y}_\delta \sin \hat{\phi}_v \\ \hat{y}_v + \hat{x}_\delta \sin \hat{\phi}_v + \hat{y}_\delta \cos \hat{\phi}_v \\ \hat{\phi}_v + \hat{\phi}_\delta \\ \hat{x}_m \end{pmatrix}$$

$$P_a^- = \nabla f_{x_a} P_a \nabla f_{x_a}^T + \nabla f_{x_\delta} P_\delta \nabla f_{x_\delta}^T$$

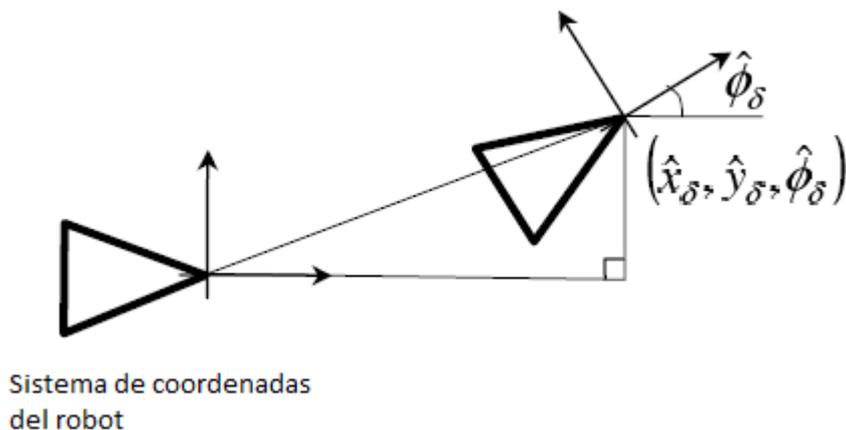


Figura 3.3: Cambio en el vector de la pose. Existen muchos algoritmos para calcular la pose una vez movido el robot, aunque los más utilizados son los relacionados con asociación de datos.

donde las jacobianas ∇f_{x_a} ∇f_{x_δ} son definidas como

$$\begin{aligned}\nabla f_{x_a} &= \frac{\partial f}{\partial x_a} \Big|_{(x_a, x_\delta)} = \begin{pmatrix} \nabla g_{x_v} & 0_{vm} \\ 0_{vm}^T & I_m \end{pmatrix} \Big|_{(x_a, x_\delta)} \\ &= \frac{\partial g}{\partial x_\delta} \Big|_{(x_a, x_\delta)} = \begin{pmatrix} \nabla g_{x_\delta} \\ 0_{vm} \end{pmatrix}\end{aligned}$$

y las jacobianas ∇g_{x_v} y ∇g_{x_δ} son como las siguientes

$$\begin{aligned}\nabla g_{x_v} &= \frac{\partial g}{\partial x_v} \Big|_{(x_a, x_\delta)} = \begin{pmatrix} 1 & 0 & -\hat{x}_\delta \sin \hat{\phi}_v - \hat{y}_\delta \cos \hat{\phi}_v \\ 0 & 1 & \hat{x}_\delta \cos \hat{\phi}_v - \hat{y}_\delta \sin \hat{\phi}_v \\ 0 & 0 & 1 \end{pmatrix} \\ \nabla g_{x_\delta} &= \frac{\partial g}{\partial x_\delta} \Big|_{(x_a, x_\delta)} = \begin{pmatrix} \cos \hat{\phi}_v & -\sin \hat{\phi}_v & 0 \\ \sin \hat{\phi}_v & \cos \hat{\phi}_v & 0 \\ 0 & 0 & 1 \end{pmatrix}\end{aligned}$$

Como estas jacobianas afectan solamente a la porción de la matriz P_v y sus correlaciones cruzadas P_{vm} , la ecuación puede escribirse como:

$$P_a^- = \begin{pmatrix} \nabla g_{x_v} P_v \nabla g_{x_v}^T + \nabla g_{x_\delta} P_\delta \nabla g_{x_\delta}^T & \nabla g_{x_v} P_{vm} \\ (\nabla g_{x_v} P_{vm})^T & P_m \end{pmatrix} \quad (3.8)$$

3.5.3. Paso de actualización

Si un punto característico ya está almacenado en el mapa con su estimada (x_i, y_i) es observado por un sensor de barrido con una medida:

$$z = \begin{pmatrix} r \\ \theta \end{pmatrix}$$

$$R = \begin{pmatrix} \sigma_r^2 & \sigma_{r\theta}^2 \\ \sigma_{r\theta}^2 & \sigma_\theta^2 \end{pmatrix}$$

donde (r, θ) es el rango y el giro relativo al observador y R es la covarianza de la observación, entonces la información sensada relacionada al mapa seguirá la siguiente ecuación.

$$\hat{z}_i = h_i(\hat{x}_a) = \begin{pmatrix} \sqrt{(\hat{x}_i - \hat{x}_v)^2 + (\hat{y}_i - \hat{y}_v)^2} \\ \arctan\left(\frac{\hat{y}_i - \hat{y}_v}{\hat{x}_i - \hat{x}_v}\right) - \hat{\phi}_v \end{pmatrix} \quad (3.9)$$

Asumiendo la correcta asociación de datos de la observación z a la estimación de la característica del mapa (x_i, y_i) , la ganancia de Kalman puede ser determinada como:

$$\nu_i = z - h_i(\hat{x}_a^-)$$

$$S_i = \nabla h_{x_a} P_a^- \nabla h_{x_a}^T + R$$

$$W_i = P_a^- \nabla h_{x_a}^- S_i^-$$

donde la jacobiana ∇h_{x_a} viene dada por

$$\nabla h_{x_a} = \frac{\partial h_i}{\partial x_a} \Big|_{\hat{x}_a^-} = \begin{pmatrix} -\frac{\Delta x}{d} & -\frac{\Delta y}{d} & 0 & 0 & \dots & 0 & \frac{\Delta x}{d} & \frac{\Delta y}{d} & 0 & \dots & 0 \\ \frac{\Delta y}{d^2} & -\frac{\Delta x}{d^2} & -1 & 0 & \dots & 0 & -\frac{\Delta y}{d^2} & \frac{\Delta x}{d^2} & 0 & \dots & 0 \end{pmatrix}$$

$$\Delta x = \hat{x}_i - \hat{x}_v$$

$$\Delta y = \hat{y}_i - \hat{y}_v$$

$$d = \sqrt{(\hat{x}_i - \hat{x}_v)^2 + (\hat{y}_i - \hat{y}_v)^2}$$

Para mapas SLAM de gran número de características, la jacobiana ∇H_{x_a} consiste principalmente de términos nulos permitiendo unos cálculos eficientes de las ecuaciones para el cálculo de S_i y W_i . Los términos distintos de cero alinean las posiciones de los estados del vehículo con las características observadas (\hat{z}_i, \hat{y}_i) en el vector de estado aumentado. La estimación a posteriori puede ser calculada con las siguientes ecuaciones de actualización

$$\hat{x}_a^+ = \hat{x}_a^- + W_i \nu_i$$

$$P_a^+ = P_a^- - W_i S_i W_i^T$$

El modelo de observación relaciona la estimación de características a la estimación de la pose del vehículo y sirve para reducir la incertidumbre de ambos. A través de la correlación de la pose del vehículo, las características del mapa pasan a estar correladas unas con otras y estas correlaciones van incrementándose monótonamente hasta que sus ubicaciones pasan a ser perfectamente conocidas. Como nota práctica, si se obtienen varias observaciones de una vez, se puede obtener una actualización más precisa. La razón por la que sucede esto es porque el EKF implementa corrección error linealizado y la actualización tiende a ser empujada en la dirección correcta cuanto el vector v de innovación tiene más errores de observación de una sola vez. Esta diferencia se accentúa cuando la pose del vehículo es muy incierta antes del paso de actualización. Sin embargo, una desventaja de actualizar todos los datos de golpe es que el procesado que se requiere en la matriz de covarianza de la innovación S requiere computaciones del orden n^3 donde n es el numero de características del paquete.

3.5.4. Aumento de estado

Cuando el entorno es explorado, nuevas características van siendo observadas y tienen que ser añadidas al mapa almacenado. Un método para iniciar las nuevas características se explicará a continuación. Primero, el vector de estado y la matriz de covarianza son aumentadas con los valores polar de z y su covarianza R , medidas relativamente al observador.

$$x_{aug}^{\hat{}} = \begin{pmatrix} \hat{x}_a \\ z \end{pmatrix}$$

$$P_{aug} = \begin{pmatrix} P_v & P_{vm} & 0 \\ P_{vm}^T & P_m & 0 \\ 0 & 0 & R \end{pmatrix}$$

La función g_i es derivada para convertir las observaciones polares de z a coordenadas globales del sistema. Esta transformación es función de la nueva observación y la pose actual del vehículo.

$$g_i(x_v, z) = \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} x_v + r \cos(\theta + \phi_v) \\ y_v + r \sin(\theta + \phi_v) \end{pmatrix} \quad (3.10)$$

El estado aumentado puede ser inicializado a los valores correctos aplicando una transformación linealizada por la función f_i de la siguiente manera.

$$\hat{x}_a^+ = f_i(x_{aug}^{\hat{}}) = \begin{pmatrix} \hat{x}_a \\ g_i(\hat{x}_v, z) \end{pmatrix}$$

$$P_a^+ = \nabla f_{x_{aug}} P_{aug} \nabla f_{x_{aug}}^T$$

donde la jacobiana $\nabla f_{x_{aug}}$ viene dad por

$$\nabla f_{x_{aug}} = \frac{\partial f_i}{\partial x_{aug}}|_{x_{aug}} = \begin{pmatrix} I_v & 0 & 0 \\ 0 & I_m & 0 \\ \nabla g_{x_v} & 0 & \nabla g_z \end{pmatrix} \quad (3.11)$$

y las jacobianas ∇g_{x_v} y ∇g_z son las siguientes

$$\begin{aligned} \nabla g_{x_v} &= \frac{\partial g_i}{\partial x_v}|_{(x_v, z)} = \begin{pmatrix} 1 & 0 & -r \sin(\theta + \hat{\phi}_v) \\ 0 & 1 & r \cos(\theta + \hat{\phi}_v) \end{pmatrix} \\ \nabla g_z &= \frac{\partial g_i}{\partial z}|_{(x_v, z)} = \begin{pmatrix} \cos(\theta + \hat{\phi}_v) & -r \sin(\theta + \hat{\phi}_v) \\ \cos(\theta + \hat{\phi}_v) & r \cos(\theta + \hat{\phi}_v) \end{pmatrix} \end{aligned}$$

La matriz de multiplicación en la ecuación en la que se calcula P_a^+ requiere computación del orden n^3 donde n es el número de características del mapa. Dado a la sharpness de la matriz jacobiana, es una implementación mucho más eficiente es posible con la nueva característica y con correlaciones cruzadas con el resto del mapa:

$$P_a^+ = \begin{pmatrix} P_v & P_{vm} & P_v \nabla g_{x_v}^T \\ P_{vm}^T & P_m & P_{vm}^T \nabla g_{x_v}^T \\ \nabla g_{x_v} & \nabla g_{x_v} P_{vm} & \nabla g_{x_v} P_v \nabla g_{x_v}^T + \nabla g_z R \nabla g_z^T \end{pmatrix} \quad (3.12)$$

Eliminar características del mapa SLAM es sencillo. Los elementos son simplemente borrados del vector de estado y las filas y columnas asociadas a la matriz de covarianza son eliminadas[1].

Capítulo 4

Asociación de datos

El problema de la asociación de datos se refiere al concepto de relacionar los estados de cada media o paquete de medidas con el original. Esto tiene muchas aplicaciones cuando las asociaciones son conocidas a priori o el problema de elegir el dato correcto se hace trivial. Sin embargo, para robots autónomos que intentan resolver el problema del SLAM esto es un aspecto clave. De hecho, es el más difícil de los aspectos que conciernen el SLAM, y es un aspecto esencial para poder construir un mapa correcto y una asociación incorrecta puede invalidar todo el proceso

La mayoría de las implementaciones de SLAM basado en características modelan los puntos de referencia con figuras geométricas simples, tales como puntos o líneas, lo que significa que los mapas están representados por una serie de figuras idénticas, solamente diferenciables por su localización. Estas representaciones derivan por si mismas en métodos de asociación de datos desarrollados para el seguimiento de objetivos donde la correspondencia es restringida por "información geométrica estadística". Sin embargo, los problemas generales de seguimiento de objetivos asumen que los objetivos se mueven independientemente y las asociaciones observación-a-objetivo son procesados sin considerar posibles correlaciones en el movimiento de los objetivos. La implicación de SLAM es que cada observación es procesada de manera individual y la asociación basada en incertidumbres en la medida, localización de puntos de referencia y pose del robot. Es muy importante tener en cuenta que si, la pose del robot es muy incierta entonces la asociación de datos se hace de forma bastante mala.

En este capítulo se tratará de hacer una pequeña introducción sobre como está el estado del arte en este aspecto. Las técnicas existentes se pueden agrupar en dos grandes

grupos, la asociación de medidas individuales y la asociación de paquetes de datos. Estas técnicas llevan un transfondo matemático muy denso, y el éxito de sus implementaciones dependen de multitud de parámetros, como es la incertidumbre, la forma de adquirir datos, etc, y suelen ser particulares para cada tipo de SLAM.

4.1. Asociación de medidas individuales

La forma más simple de asociar datos es tener que asociar un punto con la característica apropiada del entorno. Esta forma de resolver este problema, aparece muy comúnmente en artículos basados en el seguimiento de objetivos, que es de donde muchas de estas ideas nacieron.

4.1.1. Máxima verosimilitud

Una de las ideas más básicas y simples para realizar una asociación de datos es considerar la verosimilitud de las medidas. Esto se hace calculando la verosimilitud de que tiene cada punto de referencia con el robot y se asocia con la medida individual con la que se esté trabajando. Esta técnica se conoce como "Nearest-Neighbor data association", en español asociación de datos del vecino más próximo.

En general calcular el vecino más próximo puede ser realizado por cualquier distribución de la probabilidad, mientras que esta probabilidad sea posible calcularla para cada medida. Para todos los casos que se traten en este capítulo se tienen en cuenta los siguientes aspectos.

- Las medidas reales tomadas en en momentos actual tienen una distribución normal, por ejemplo el ruido de las medidas es gaussiano.
- El conocimiento a priori de las características del ruido de las medidas es conocido en términos de la desviación típica y de la media.

Con esto la asociación de datos se realiza tomando la verosimilitud máxima

$$e_l = \operatorname{argmax}(f_{ij}) \quad (4.1)$$

$$f_{ij} = \frac{1}{(2\pi)^{n/2} \sqrt{|S_j|}} \exp\left(-\frac{1}{2} \nu_{ij}^T S_j^{-1} \nu_{ij}\right) \quad (4.2)$$

La variable $e_l = i, j$ es definida como la l -ésima asociación para el dato i y la característica j . La variable ν_{ij} es la innovación para la pareja i, j , S_j es la matriz de innovación de la

covarianza, n es la dimension del vector de innovación que es definido como la diferencia entre la medida real y la estimada que da la asociación $e_l = i, j$.

Esto mismo se puede hacer tomando la máxima verosimilitud logarítmica, o alternativamente la mínima verosimilitud logarítmica n_k , conocida también como distancia normalizada.

$$\ln(f_{ij}) = -\frac{1}{2}\nu_{ij}^T S_j^{-1} \nu_{ij} - \ln\left(\frac{1}{2\pi^{n/2} \sqrt{|S_j|}}\right) \quad (4.3)$$

$$e_l = \operatorname{argmax}(\ln(f_{ij})) \quad (4.4)$$

$$N_{ij} = \nu_{ij}^T S_j^{-1} \nu_{ij} + \ln|S_j| \quad (4.5)$$

$$e_l = \operatorname{argmin}_j(N_{ij}) \quad (4.6)$$

A la función de máxima verosimilitud se puede incluir fácilmente la posibilidad bien de rechazar medidas falsas o erróneas, o bien permitir la posibilidad de tomar medidas de un punto de referencia antes desconocido. Si solamente una o la otra va a ser permitida, un solo corte puede ser usado como verosimilitud de una medida falsa o de un punto de referencia anteriormente desconocido.

Esto se puede hacer ad hoc mirando los errores estadísticos de las medidas y usando esta información para calcular cortes razonables que puedan constituir la medida de forma que no sea consistente con el conocimiento del entorno. Alternativamente, un método más riguroso puede ser usando el hecho que el exponente de la verosimilitud siga una distribución χ^2

4.1.2. Compatibilidad Individual

El método de la compatibilidad individual por sí mismo no es un método en el cual las asociaciones puedan ser elegidas. En vez de esto estas asociaciones estadísticamente incompatibles pueden ser eliminados de esta consideración. Este método es conocido también como reducción de la ambigüedad en contraposición de la gestión de la ambigüedad que proponen métodos como el de máxima verosimilitud.

Esto se puede hacer notando que el exponente de la función de verosimilitud que describe la distribución sigue una χ^2 .

$$M_{ij} = \nu_{ij}^T S_j^{-1} \nu_{ij} \quad (4.7)$$

Donde M_{ij} tiene una distribución χ^2 y es llamada la distancia de *Mahalanobis*. La compatibilidad individual viene dada por la siguiente inecuación:

$$M_{ij} < \gamma_n \quad (4.8)$$

La cantidad γ_n es determinada a partir de dos parámetros, n , la dimensión del vector de innovación y un parámetro libre que define el porcentaje esperado de asociaciones correctas. Lo normal es usar unos parámetros del 95 % .

4.1.3. Combinación de compatibilidad individual y máxima verosimilitud

Otra posibilidad para asociar medidas individuales es usar las dos técnicas anteriormente explicadas simultáneamente. Esto se hace primero procesando las medidas vía compatibilidad individual y luego eliminando aquellas asociaciones que fallen en la compatibilidad individual. Luego, con las asociaciones restantes, se eligen las que tengan mayor verosimilitud. La única diferencia en resultados de esta técnica con la de la máxima verosimilitud es que permite la eliminación previa de medidas falsas. En otras palabras habrá casos en los que este método no tome ninguna asociación como válida, y por lo tanto ignorará las medidas al tratarlas como falsas.

4.2. Asociación de paquetes de datos

El caso de asociación de paquetes de datos es muy similar al de las medidas individuales y muchos de los algoritmos desarrollados para ello pueden ser utilizados también para el caso de medidas individuales. La asociación de datos se refiere a la situación en la cual los aparatos de medida usados, o la dinámica de la situación, hacen posible tratar un paquete de medidas todas simultáneamente. Este caso se da cuando se utilizan aparatos de medida como cámaras, o sensores láser de barrido. Otro caso es que la frecuencia de muestreo sea muchísimo mayor que la dinámica del vehículo, de manera que el movimiento entre la toma del primer dato y el último pueda ser ignorado.

Tratar el problema de la asociación de datos como un paquete tiene muchas ventajas, que usadas correctamente, pueden facilitar esta labor a la vez que permiten una mejor asociación de estos. La primera es la exclusión por aglutinamiento, que quiere decir que sólo puede haber una medida de un paquete asociada a un solo punto de referencia. Si esta

condición es aplicada correctamente, permite purgar gran cantidad de medidas que podrían conllevar a realizar asociaciones erróneas. La otra ventaja es que con estas técnicas se pueden comparar todas las parejas de asociaciones con la asociación de todos los datos.

En este aspecto existen muchas técnicas, que se explican en la bibliografía, pero caben destacar: *joint compatibility branch and bound*, *maximum common subgraph*, *combined constraint data association*, etc. Pero estos temas se exceden de los objetivos del proyecto, por lo que no se profundizará en ellos.

Capítulo 5

Scan-Matching

Cómo ya se comentó en la introducción, no siempre es fácil sacar características geométricas de un entorno, y cuando esto ocurre hacer una asociación de datos correcta es muy complicado, como en el ejemplo de la figura 5.1. En estos casos a veces es muy útil asociar datos directamente de la observación con el mapa. Esta técnica es muy útil también en entornos donde funciona bien la asociación de datos, pues incorpora toda la información disponible en los sensores (no rechaza datos sin modelar).

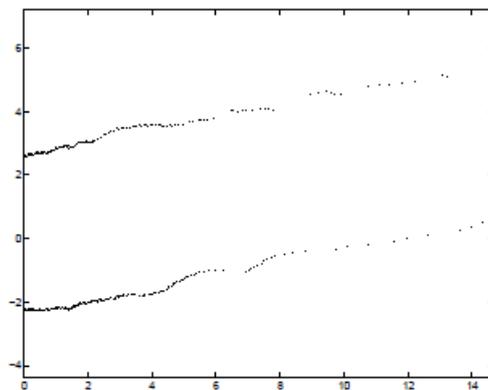


Figura 5.1: Datos en bruto de un túnel de una mina. En este entorno es complicado extraer características geométricas. En este tipo de entornos resultan muy útiles técnicas de scan-matching.

Existen un gran número de técnicas en este aspecto como se verá en este capítulo, pero la mayoría no tienen una derivación probabilística. Por ello en esta sección se explicará en detalle un método que proporciona la incertidumbre relativa de la pose.

5.1. Definición del problema

5.1.1. La estimación de la pose mediante el alineamiento de barridos

Un barrido de rangos es una lista correspondientes a la intersección de un haz láser con objetos del entorno del robot. El haz láser va rotando en un plano horizontal y emana de un sensor montado en el robot. Este barrido de rangos es una rodaja 2D del entorno. Los puntos de este barrido pueden ser representados en coordenadas polares cuyo origen sería la posición del sensor, y el sentido de giro vendría especificado por el giro del sensor. Cada punto es representado por la dirección del haz láser y la medida de la distancia a lo largo de esa dirección. En este trabajo se referirá a $(O(x,y),\theta_0)$ como la pose del escaneo, donde O es la posición del sensor en un sistema de coordenadas global del sistema y θ_0 es la orientación por defecto del sensor.

Suponga se que el robot empieza en una pose P_{ref} (a partir de ahora será la pose de referencia) y que hace un escaneo (llámese escaneo de referencia S_{ref}). Ahora el robot se mueve en un entorno estático hasta una nueva pose P_{nuevo} y hace un nuevo escaneo S_{nuevo} . La diferencia de poses entre P_{nuevo} y la pose de P_{ref} (relativo a traslación y rotación) normalmente se puede conocer de manera aproximada con los datos de la odometría. Sin embargo, en la mayoría de los casos esta información no es correcta dado el deslizamiento de las ruedas. El objetivo de esta técnica es obtener la diferencia exacta de la pose P_{nuevo} con respecto a P_{ref} , alineando los dos barridos.

El problema de cuadrar es formulado de la siguiente manera: inicialmente se asume que la pose de S_{nuevo} es P'_{nuevo} , entonces se encuentra una rotación ω y una traslación T para S_{nuevo} de forma que, después de la transformación, S_{nuevo} esté alineado con S_{ref} .

5.1.2. Criterio para alinear barridos

Un barrido es una secuencia de puntos que representan el contorno curvo de un entorno local. Dada la existencia de ruido aleatorio en el sensado y a la auto-oclusión, puede resultar imposible alinear dos barridos de forma exacta. Existen dos tipos de discrepancias entre barridos.

Naturalmente, se adoptará un criterio para alinear los barridos de forma que se bus-

cará el mejor alineamiento de la parte solapada de forma que se minimice por mínimos cuadrados el error, mientras que se ignorará el resto. Se puede formular este criterio como la minimización de la distancia medidas entre dos barridos, como función de la rotación y la traslación. Para excluir los sobrantes en las medidas se pueden aplicar estadística compleja.

5.2. Técnicas de Scan matching

En esta parte se explicará brevemente las técnicas más usadas pra realizar Scan-Matching. Se presentarán varios métodos no probabilísticos como el ICP, IMRP, o el angulo de histograma y otros que pueden ser implementados de manera probabilística.

5.2.1. Alineado basado en la correspondencia de puntos

Introducción

Este es un algoritmo iterativo y está basado en asociar correspondencia punto a punto. La idea de este método es la siguiente. Para cada punto P_i de S_{nuevo} , se usa una regla muy simple (que es independiente de la rotación y la traslación actuales) para determinar un punto correspondiente P'_i de S_{ref} . Entonces de todas los pares correspondientes de puntos, se puede calcular una solución por mínimos cuadrados relativa rotación y traslación. Esta solución es aplicada para reducir el error en la pose entre dos barridos. Este proceso se repite hasta que converge.

La solución de mínimos cuadrados deriva de minimizar la siguiente función de la distancia que es definida para n pares de puntos correspondientes:

$$E_{dist} = (\omega, T) = \sum_{i=1}^n |R_{\omega}P_i + T - P'_i|^2 \quad (5.1)$$

La idea central de este algoritmo es definir una regla sensible para determinar correspondencias sin necesidad de conocer la rotación ni la traslación actuales.

Iterative Closest Point (ICP)

Es una de las técnicas mas usadas. Consiste en escoger el punto más cercano del modelo para un punto dado. A esta regla se la llama regla del punto más cercano, *Closest-point rule* en inglés. La figura 5.2 muestra un ejemplo de como encontrar correspondencias

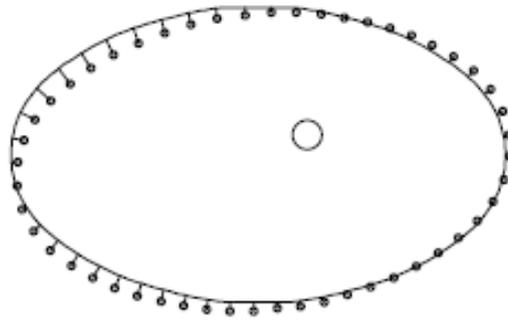


Figura 5.2: Resultados de puntos más cercanos usando ICP. Se puede ver en la imagen que no se tiene mucha información sobre la rotación.

para un juego de datos elíptico, usando esta técnica. Besl y McKay describieron un algoritmo de uso general llamado ICP iterative closest point. Este algoritmo siempre converge monóticamente a un mínimo local con respecto a la ecuación de la distancia que se explicó en el apartado anterior. Ese algoritmo funciona muy bien y muy rápidamente sobre todo cuando existe poca rotación.

Un inconveniente del algoritmo ICP es que converge muy lentamente. Especialmente, cuando el modelo es curvo, las correspondencias encontradas por la regla del punto más cercana pueden contener poca información sobre la rotación. Como se ve en la figura, los vectores que unen los pares correspondientes es difícil saber que dirección llevan y tienden a anularse unos a otros cuando se intenta resolver la rotación. Independientemente del modelo, la velocidad de convergencia del algoritmo es siempre lenta cuando la función se aproxima a un mínimo local.

Para acelerar el algoritmo ICP, Besl y McKay usaron un método de búsqueda de líneas para determinar heurísticamente la transformación de variables basado en sus valores en dos o tres iteraciones. Aunque esto mejora la velocidad de convergencia en zonas próximas a un mínimo local, el problema de que los resultados muestren poca información sobre la rotación siguió sin ser resuelto. En definitiva el tiempo de convergencia de este algoritmo es lento, para poder aplicar el método de búsqueda de líneas hay que hacer unas asunciones en la ecuación de mínimos cuadrados de la distancia.

Recientemente se ha estudiado variantes eficientes del algoritmo ICP. En concreto Szymon Rusinkiewicz, ha conseguido obtener una versión de ICP de alta velocidad, si se quiere más información sobre como ha sido implementado puede mirar la bibliografía [6].

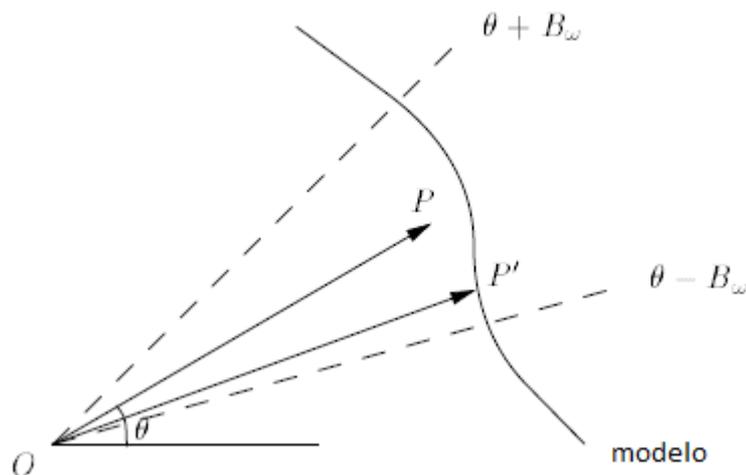


Figura 5.3: Regla para matching-range-point. Para un punto P , el correspondiente P' en el barrido está en el sector y es punto más cercano a P

Matching Range Point rule

A continuación se propone una regla que encuentra correspondencias de tal forma que se obtiene información de la rotación.

Considérese un punto P y su correspondiente $P' = R_\omega P + T$. Si se ignorase la traslación, se obtendría entonces que $|P'| \approx |P|$. Por otro lado, el ángulo polar $\hat{\theta}$ de P' está relacionado por $\hat{\theta} \approx \theta + \omega$. Esto implica que la correspondencia de P bajo una rotación es un punto que tiene el mismo rango polar que P , y los ángulos polares difieren en el ángulo de rotación ω . Ahora en el caso de una pequeña traslación, todavía se puede esperar que P' siga teniendo el mismo rango que P , luego se puede esperar buenas correspondencias entre P y P' , y esto da muchísima información sobre la rotación ω .

Para asegurarse de que la regla encuentra una única solución, sólo se busca al matching range point en una región local cerca de P . Supóngase que se puede estimar un límite B_ω para la rotación ω , por ejemplo, $|\omega| \leq B_\omega$. Se tiene pues que $\hat{\theta} \in [\theta - B_\omega, \theta + B_\omega]$. Esto significa que P' debería estar en la región $\theta \pm B_\omega$. De esta forma, se puede definir que: El método para cuadrar datos basado en el rango como, para un punto P , su correspondiente punto P' en el modelo satisface la ley $|\hat{\theta} - \theta| \leq B_\omega$ y $|P'|$ es el punto más cercano de $|P|$. En la figura 5.3 se muestra la regla ilustrada.

En la figura 5.4 se puede ver un ejemplo usando esta técnica para buscar correspon-

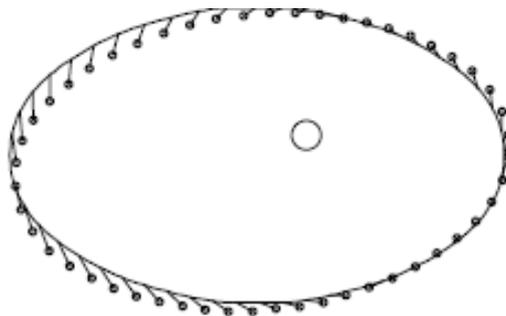


Figura 5.4: Puntos más cercanos usando la técnica IMRP. Se puede observar que se obtiene más información sobre el giro que con el algoritmo ICP

dencias en un modelo elíptico. Se puede observar claramente que los vectores que unen los pares de datos facilitan información sobre la dirección de la rotación. De esta forma, la solución por mínimos cuadrados debiera ser una buena aproximación para la transformación real, especialmente en el componente de la rotación [6].

Basándose en esta idea, Feng Lu y Evangelos Milios, diseñaron un algoritmo iterativo llamado IMRP (*Iterative Matching Range Point*). En este algoritmo, el parámetro B_ω controla el tamaño de la región en la cual se buscará la correspondencia y también la rotación máxima que sería posible resolver en un sola iteración. Según esto lo mejor sería escoger un B_ω de forma que sea próximo al residuo de la rotación en cada iteración. Empíricamente se puede generar B_ω usando una función exponencial decreciente de la forma: $B_\omega(t) = B_\omega(0)e^{-\alpha t}$ Es posible también usar la solución de mínimos cuadrados de ω en la iteración actual para determinar el valor de B_ω en la siguiente iteración.

Angulo de histograma

Este método solo se puede usar con sensores de barrido 2-D y con precisiones del láser. Esto es el sensor tiene que devolver unos datos de medición espaciados exactamente el mismo ángulo entre unos y otros. Además cada barrido tiene que cubrir 360, lo que quiere decir que tiene que ser una secuencia de medidas circulares. La esencia de este algoritmo es la siguiente. Cada medida (r_i, θ_i) de cada barrido, es proyectada a un sistema de coordenadas cartesiano centrado en el observador. Dadas 2 medidas consecutivas, $v_i = (x_i, y_i)$ y $v_{i+1} = (x_{i+1}, y_{i+1})$ y el ángulo entre ellos será:

$$\alpha_i = \arctan\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) \quad (5.2)$$

El histograma de ángulos es generado a partir de los ángulos de pares de datos adyacentes de cada barrido. La forma de este histograma es invariable para rotación pura, que

aparece como un desfase. Así la rotación del sensor puede ser hallada como correlación cruzada de dos histogramas. La relación entre el desfase y la rotación del sensor es válida incluso si existe una pequeña cantidad de translación por parte del sensor. Así pues, el cambio en la pose del sensor es hallado primero rotando la observación el ángulo de desfase, y luego encontrando el desfase en la translación usando histogramas de x e y [1].

Este método no presenta ninguna ventaja sustancial sobre el ICP. Tampoco es más preciso, ni eficiente, ni tampoco más fácil de implementar. Quizás la única ventaja es que no requiere asociación explícita punto a punto.

Correlacion de matriz de ocupación

Las matrices de ocupación como ya se explicó en el segundo capítulo, usan correlación de barridos para introducir las observaciones en el mapa. El método para incluir estos datos es correlación 2-D, que se implementa de manera similar a como se hace *template matching* en el procesado de imágenes. Esto es, la celda de la observación es movida por una pose dada y el valor de la correlación para esa pose es calculada por (i) multiplicando por si mismas las celdas que se superponen, y (ii) sumando este resultado a cada celda de observación. Nótese que cuando el observador varía su pose en x, y o θ , el espacio de búsqueda pasa a ser tridimensional. Esto es las 2 matrices no se solapan limpiamente y es necesario usar interpolaciones. También el espacio de búsqueda es mucho mayor que para 2-D, y la correlación cruzada puede ser no factible sin establecer límites a priori. También depende de la relación entre eficiencia y granualidad como se explicó en el capítulo 2.

Las matrices de ocupación representan la información del sensor en un espacio de parámetros multidimensional, donde cada celda tiene una variable de estado distinta entre 0 y 1. Los valores y la correlación de estos estados son actualizados de una manera no bayesiana de acuerdo a los modelos geométricos del sensor, como por ejemplo el modelo del sonar. Esto es, que el sensor es modelado en el espacio, en términos de probabilidad de ocupación, en vez de directamente relacionar la incertidumbre de la medida al estado de cada parámetro de cada celda. Esto significa que la relación entre la incertidumbre del sensor y la incertidumbre de la correlación cruzada no están definidos con precisión. La correlación cruzada si que puede seguir siendo probabilística pero no asociada al modelo del sensor en un sentido Bayesiano.

Métodos probabilísticos

Recientemente han aparecido técnicas de scan-matching que utilizan principios de probabilidad, diciendo que la correlación viene dada por una función de verosimilitud "*likelihood function*" derivada de un modelo estocástico de la observación. Normalmente estos métodos son presentados como algoritmos de máxima verosimilitud, que intentan encontrar el valor que maximiza la función de verosimilitud, pero pocas propuestas demuestran una estimación de probabilidad de densidad, combinando con funciones de verosimilitud con información anterior para calcular a posteriori la PDF (*probability density function*) de la pose.

Aparecen principalmente dos métodos con correlaciones de verosimilitud máximas. El primero es parecido al ICP, iterativamente buscando asociarse con el vecino más cercano para cada punto en los barridos de observación. Cada asociación tiene un valor de verosimilitud, y el producto de estas verosimilitudes da como resultado la verosimilitud del barrido. La máxima verosimilitud de la pose puede ser encontrada muestreando valores en la solución de estado, normalmente empleando un algoritmo de optimización aglutinado, en inglés *greedy* para converger en un máximo local. El problema de estas técnicas basadas en ICP es que los modelos de verosimilitud no representan con mucha precisión el modelo de la incertidumbre del sensor.

La segunda técnica de correlación de verosimilitud máxima es la técnica basada en matrices de ocupación. Sin embargo, la diferencia es que en este caso se trabajaría con matrices de probabilidad en vez de matrices de ocupación. Mientras las matrices de ocupación 2-D definen un espacio de parámetros elevado, de tamaño igual al número de celdas, las matrices de probabilidad simplemente representan un espacio de parámetros 2-D. La principal debilidad de esta técnica es que se discretiza la función de distribución, lo que conlleva pérdida de información. En algunos casos, una mejor PDF se puede conseguir usando, por ejemplo, wavelets, suma de gaussianas, etc. En este trabajo se centrará la atención en la técnica como suma de gaussianas.

5.2.2. Representación probabilística de los datos

Para realizar un Scan-Matching probabilístico, es necesario definir antes un espacio de parámetros adecuado para los datos, y definir también una función de verosimilitud bayesiana para la pose relativa entre dos barridos.

En esta sección, se usarán barridos 2-D del entorno con n medidas cada uno de ellos. Cada medida del láser observa la presencia de una superficie en su envolvente y tendrá un alcance máximo. La posición actual de una superficie tiene una cierta incertidumbre, que es modelada por una PDF gaussiana. Entonces, para un paquete de n medidas, hay un lote de n gaussianas que describen la posición de una superficie discreta. Estas superficies son distintas unas de otras pero pueden estar correladas debido a su proximidad, la suavidad del entorno o por la apertura del sensor.

Existen dos formas de parametrización en el espacio de estados. La primera es una proyección de un espacio de medidas 2-D a un espacio cartesiano 2-D que representa las coordenadas del mundo. El segundo de mayor orden, es que las medidas son mapeadas a un espacio no cartesiano de dimensión mayor que dos.

Representación como PDF bidimensional

Una PDF 2-D se genera a partir de las medidas de un scan transformándolo a un espacio gaussiano centrado en el sensor, y acumulando estas distribuciones escalandolas y sumandolas. La suma de gaussianas resultante define la distribución de la incertidumbre en el espacio paramétrico cartesiano 2-D.

El espacio de parámetros 2-D implica que el barrido representa un solo punto como objetivo. Esto está implícito en su dimensionalidad, la PDF define la incertidumbre como un vector aleatorio (x, y) para el cual estos pueden existir con un solo valor (x_t, y_t) .

Este modelo tiene las siguientes ventajas. Primero, que es general, esto quiere decir que la conversión entre incertidumbres de medidas a PDF en el espacio cartesiano no involucra modelos geométricos. Segundo, como el modelo involucra una sola entidad no existe el problema de la asociación de datos. El alineamiento es determinado por la correlación cruzada de dos PDF de barridos. Tercero, la correlación cruzada de la suma de dos gaussianas puede ser calculado fácilmente.

Sin embargo, la representación en el espacio de parámetros 2-D tiene dos dificultades importantes. La primera, es que el modelo de punto objetivo no es una representación realística de los datos. La representación en el espacio 2-D, efectivamente convierte la interpretación de una serie de medidas de superficie discretas a unas posibles localizaciones. La distribución de una gaussiana dice en definitiva que *un objetivo podría estar*

aquí mientras que otra diría *o aquí*.

Mientras que el modelo 2-D tiene muchas imperfecciones, no significa que sea del todo incorrecto. Por ejemplo, para una sola medida, tanto la interpretación real como la artificial dicen que "hay un punto en algún lugar de esta distribución". Más importante es que dados dos barridos, por lo menos compartan algo en común, luego usando la correlación de los barridos en la representación PDF 2-D, se obtendrá una función de verosimilitud de la pose bastante consistente. Esta propiedad aparece por la forma en que es calculada la función de verosimilitud, que es gracias a la correlación cruzada de dos barridos, que dará un resultado distinto de cero para la pose aunque solamente haya información en común de una sola medida en cada barrido.

El modelo del espacio de parámetros 2-D satisface 4 condiciones:

1. Generalidad: La transformación de medidas al espacio de parámetros no debe requerir modelos restrictivos, dado que es la razón por la cual se elige scan-matching frente a otras técnicas.
2. Asociación de datos: Debería evitarse los umbrales de la asociación de datos.
3. Precisión: La incertidumbre la pose resultante debería ser mínima, y consistente en el sentido que la función de verosimilitud para la pose real debería ser distinta de cero.
4. Tratable: El scan-matching debería de ser posible de realizar en tiempo real.

Alternativas de mayores dimensiones

Si el espacio de parámetros 2-D no es suficiente para describir adecuadamente el barrido, se necesita un espacio de mayores dimensiones. Existen tres posibilidades: parámetros basados en características, parámetros de medida y matrices de ocupación.

El espacio basado en el espacio de parámetros es obtenido especificando modelos de características, agrupando y clasificando las medidas del sensor de acuerdo con estos modelos. Donde sea posible, los modelos de características transforman los datos de las medidas al espacio de características $[f_1, \dots, f_k]^T$, y cada nueva característica conlleva un aumento de estado. Una posible restricción en esta representación es que para hacer la transformación hay que aplicar modelos restrictivos, que desechan parte de la información y pueden no

ser adecuados en ciertos entornos[4].

5.2.3. Representación Bayesiana de la función de verosimilitud para el modelo punto-objetivo (*point-target*)

En esta sección se va a deducir la función de verosimilitud para scan-matching basado en el modelo de punto-objetivo. Dadas dos PDF de barridos, cada una representado un solo objetivo, la función de verosimilitud para su pose relativa es calculada a través de su correlación cruzada.

La correlación de barridos es la alineación de barridos de una observación a un mapa, que contiene un sólo punto de referencia. Como advertencia, una vez que hecha la alineación del primer barrido, el mapa y la pose del robot pasan a estar correladas y no deberían ser tratados como independientes cuando se alineen las siguientes observaciones[1].

Notación

En esta sección se explicará la notación que se usará a continuación. Dado un vector aleatorio 2-D $x \in \chi$, su PDF es ya definida por la función de distribución $f_x(x, y)$. Esta función representa la PDF de (x, y) las variables escalares x e y son parámetros independientes en el espacio χ . Si dos vectores aleatorios a y b están concatenados como $[a, b]^T$, su función de distribución viene dada por $f_a b$. Por ejemplo, si a y b son 2-D y 1-D respectivamente, será $f_{ab}(x, y, z)$ [1].

El estado en un momento k es la pose del robot $x_k = [x_k, y_k, \phi_k]^T$. La función de verosimilitud se definirá como $\Lambda(x_k) = f_{x_k}(x, y, \phi)$.

Localización de un robot en un plano

En un plano, la pose del robot se compone de tres variables (x_k, y_k, ϕ_k) . Sin embargo, dado un sólo punto de referencia, como en la figura 5.5, solo es posible resolver para dos de estos estados y por lo tanto es necesario fijar uno de ellos para encontrar la solución de los otros dos. En este caso, el valor de la cabeza se asume que es conocida como $\phi_k = \phi_O$. La explicación de a continuación indica que, para cada valor fijado ϕ_k , la función de verosimilitud para la pose del robot viene dada por la correlación cruzada de la observación y el

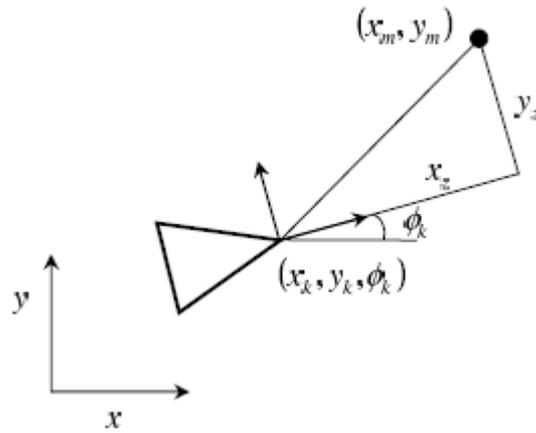


Figura 5.5: Localización del robot en dos dimensiones. Un único punto de referencia es guardado en el mapa como (x_m, y_m) , y observado por el robot por la medida (x_z, y_z)

mapa de PDF, y indica también que esto se cumple para el caso de ϕ_k . La figura 5.5, define la localización de la referencia por $x_m = [x_m, y_m]^T$ y las medidas centradas en el robot para este punto como $x_z = [x_z, y_z]^T$. Ambos x_m y x_z son variables aleatorias con PDF reorientados por $f_m(x, y)$ y $f_z(x, y)$ respectivamente (nótese que $f_m(0, 0)$ es la densidad del mapa en el origen de coordenadas, y f_z es la densidad de medidas en el origen de coordenadas el robot). El estado del vehículo es $x_k = [x_k, y_k]$ conociendo el ángulo de la cabeza ϕ_0 .

Dadas las PDF independientes $f_m(x, y)$ y $f_z(x, y)$ se desea encontrar la PDF del estado $f_{x_k}(x, y)$ (que es simplemente la función de verosimilitud $\Lambda(x_k)$ para un ϕ_k). El estado está relacionado a la observación por la siguiente ecuación[4].

$$x_k = x_m - \mathbf{R}x_z \tag{5.3}$$

dónde \mathbf{R} es la matriz de rotación

$$\mathbf{R} = \begin{pmatrix} \cos\phi_0 & -\sin\phi_0 \\ \sin\phi_0 & \cos\phi_0 \end{pmatrix} \tag{5.4}$$

Usando las variables auxiliares a_1 y a_2 para dobtener cuatro ecuaciones con cuatro incógnitas la ecuación 5.3 pasará a ser

$$\begin{aligned} x_k &= x_m - x_z \cos\phi_0 + y_z \sin\phi_0 \\ y_k &= y_m - x_z \sin\phi_0 + y_z \cos\phi_0 \end{aligned}$$

$$\begin{aligned} a_1 &= x_m \\ a_2 &= y_m \end{aligned} \tag{5.5}$$

que es referido con facilidad como $(x_k, y_k, a_1, a_2) = g(x_z, y_z, x_m, y_m)$ que tiene como solución única

$$\begin{aligned} x_z &= (a_1 - x_k)\cos\phi_0 + (a_2 - y_k)\sin\phi_0 \\ y_z &= -(a_1 - x_k)\sin\phi_0 + (a_2 - y_k)\cos\phi_0 \\ x_m &= a_1 \\ y_m &= a_2 \end{aligned} \tag{5.6}$$

La distribución de probabilidad para la ecuación 5.5 es obtenida de la siguiente forma(bibliografía)

$$f_{x_k} a(x_k, y_k, a_1, a_2) = \frac{f_z m(x_z, y_z, x_m, y_m)}{\text{abs}|\nabla g_z m|} \tag{5.7}$$

donde la jacobiana ∇g_z , es dada por

$$\nabla g_z m = \frac{\partial g}{\partial(x_z, y_z, a_1, a_2)} = \begin{pmatrix} -\cos\phi_0 & \sin\phi_0 & 1 & 0 \\ -\sin\phi_0 & -\cos\phi_0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{5.8}$$

El determinante de $\nabla g_z m$ es único, siempre que x_z y x_m sean únicos y la ecuación 5.7 pasa a ser

$$f_{x_k} a = f_z(x_z, y_z) f_m(x_m, y_m) \tag{5.9}$$

donde los valores de x_z, y_z, x_m e y_m son las soluciones de la ecuación 5.6. La PDF $f_{x_k}(x_k, y_k)$ es obtenida de este resultado integrando en todos los valores de (a_1, a_2) .

$$f_{x_k}(x_k, y_k) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f_z(b_1, b_2) f_m(a_1, a_2) da_1 da_2 \tag{5.10}$$

donde b_1 y b_2 son las soluciones de x_z e y_z , respectivamente.

Intuitivamente, la ecuación 5.10 es interpretada como la primera rotación de la PDF de la observación $f_z(x, y)$ por ϕ_0 y luego haciendo la correlación cruzada en dos dimensiones con la PDF del mapa. Esto se puede ver más fácilmente definiendo la rotación como $x_r = R x_z$. Claramente, x_z tiene la única solución $x_z = R^{-1} x_r$ y, siguiendo el mismo procedimiento de la ecuación 5.7 la PDF de x_r es

$$f_r(x, y) = f_z(x\cos\phi_0 + y\sin\phi_0, -x\sin\phi_0 + y\cos\phi_0) \tag{5.11}$$

de esta forma la ecuación 5.10 se puede reescribir como

$$f_{x_k}(x_k, y_k) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f_r(a_1 - x_k, b_2) f_m(a_1, a_2) da_1 da_2 \quad (5.12)$$

Este resultado demuestra que, para un valor fijado ϕ_k la función de verosimilitud $\Lambda(k)$ se calcula utilizando la correlación cruzada del mapa y de las distribuciones rotadas de las observaciones. Además, para cada ϕ_k fijada, la probabilidad total acumulada de $\Lambda(x_k)$ es uno y además la probabilidad se acumula para dos valores de ϕ_k iguales. Así de este modo, por la regla de proporcionalidad dice que un paquete de correlaciones cruzadas para todos los ϕ_k constituyen una función de verosimilitud válida en el dominio de x_k cuando ϕ_k varía.

5.2.4. Correlación de suma de gaussianas

En esta sección se tratará la conversión de un paquete de medidas a una PDF como suma de gaussianas y deducir estas ecuaciones para las correlaciones cruzadas de dos de estas PDF. El problema está enfocado para un sensor láser, que observa objetos en un espacio 2-D.

Los siguientes aspectos serán tratados para luego implementar un algoritmo scan-matching basado en la suma de gaussianas.

- Las ecuaciones para pasar los datos del sensor a un sistema cartesiano de suma de gaussianas
- Dos criterios para escalar las gaussianas de una en una dentro la suma, siendo es escalado de iguales alturas la opción preferida
- La deducción de las ecuaciones para la suma gaussiana tanto para el caso 1-D como 2-D.

Conversión a representación como suma de gaussianas

Una distribución gaussiana de n dimensiones, con una media de valor \bar{p} y una covarianza P , es definida por la siguiente ecuación.

$$g(x, \bar{p}, P) = \frac{1}{(2\pi)^{n/2} \sqrt{|P|}} \exp\left(-\frac{1}{2}(x - \bar{p})^T P^{-1}(x - \bar{p})\right) \quad (5.13)$$

La integral de esta función en el espacio R^n es uno, luego una gaussiana escalada $\alpha g(x, \bar{p}, P)$ tiene un volumen igual a α .

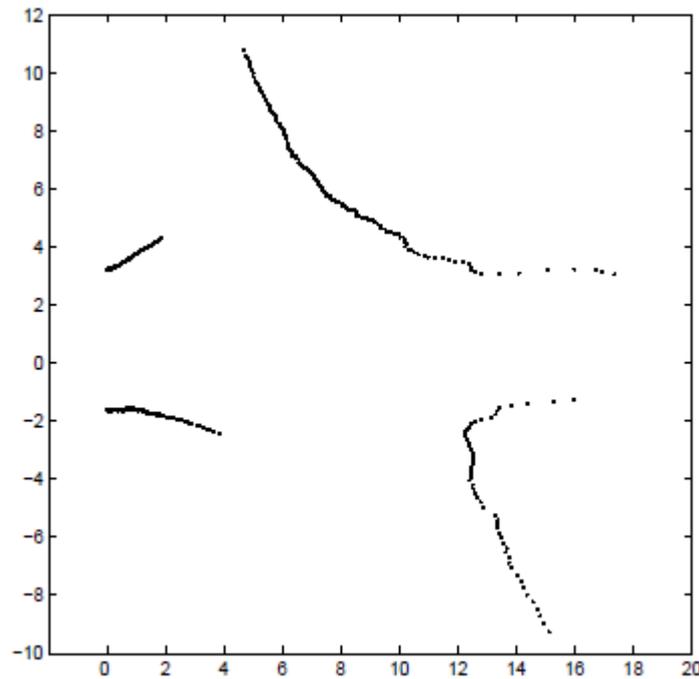


Figura 5.6: Datos obtenidos del láser sin procesar

Una PDF de la suma de gaussianas n -dimensional es definida como k gaussianas escaladas

$$G(x) = \sum_{i=1}^k \alpha_i g(x, \bar{p}_i, P_i) \quad (5.14)$$

dónde la suma de los factores de escalado α_i es igual a uno. Para el objetivo de la correlación cruzada, la probabilidad total no es relevante. Así pues en este trabajo las sumas de gaussianas no requieren normalización [4].

Modelos de escalado para la suma de gaussianas

Una forma intuitiva de sumar las gaussianas de las medidas individuales es simplemente sumarlas sin escalarlas.

$$G(x) = \sum_{i=1}^k \alpha_i g(x, \bar{p}_i, P_i) \quad (5.15)$$

Sin embargo, esta formulación da como resultado una representación del entorno curvada, por lo cual el escalado puede ser una solución óptima para este problema. Para ver esto en la figura 5.6 se puede ver los datos de un barrido láser transformados a un espacio cartesiano, y la figura 5.7 muestra la suma de PDF para este barrido, basado en la ecuación 5.15. Dado que, cada gaussiana tiene el mismo volumen, aquellos que tienen poca incertidumbre poseen valores muy elevados, indicando elevada verosimilitud. El efecto de

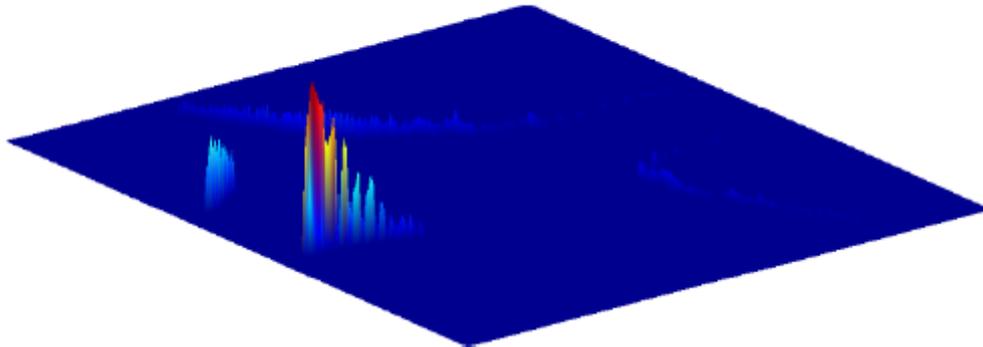


Figura 5.7: Representación de igual volumen. Los picos representan medidas con poca incertidumbre

este modelo de escalado para la correlación de los barridos, es que los puntos cercanos tienen mucha influencia en la correlación cruzada mientras que los puntos que están alejados tienen muy poca. La dificultad del escalado nace de la discrepancia entre la realidad física de los elementos sensados y la interpretación del modelo cartesiano del espacio de parámetros. El modelo de escalado de igual volumen tiene sentido si cada medida en realidad observa el objetivo, dado que cuanto más fiable sea el resultado de esta, mejor será la estimación de su localización. Sin embargo, las gaussianas individuales de la suma, tienen que ser entendidas como posibles localizaciones de los objetivos, y una medida más precisa no es más probable que una medida gruesa.

Un modelo alternativo es escalar cada gaussiana para que tengan iguales alturas, como se muestra es la siguiente ecuación.

$$G(x) = \sum_{i=1}^k \exp\left(-\frac{1}{2}(x - \bar{p}_i)^T P_i^{-1}(x - \bar{p}_i)\right) \quad (5.16)$$

En otras palabras, cada gaussiana tiene un factor de escalado $\alpha_i = (2\pi)^{n/2} \sqrt{|P_i|}$. La PDF de la suma de estas gaussianas para el barrido anterior se puede ver en la figura 5.8. Este modelo de escalado de igual altura dice que un punto tiene la misma probabilidad de estar en la distribución de una medida tanto fiable como en una no fiable, pero unas medidas más gruesas cubren más espacio, por lo cual conceden más información del entorno, entonces cuanto más anchas sean las gaussianas, tendrán pues más volumen [4].

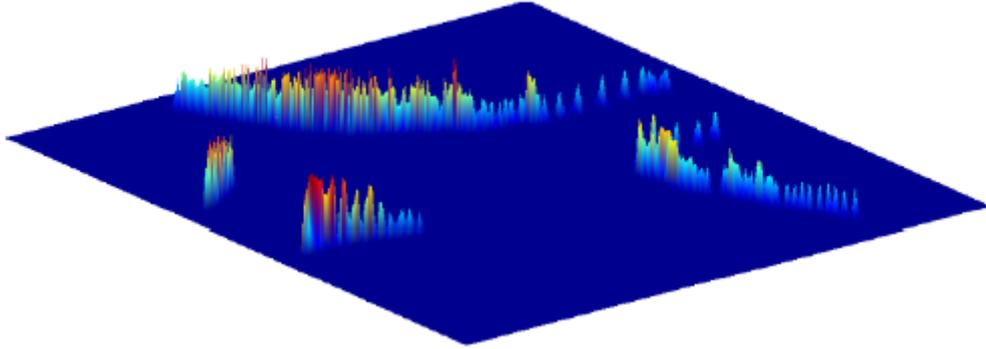


Figura 5.8: Representación de igual altura. Estipula que las medidas muy precisas contienen menos información sobre la forma del entorno, y produce mejor descripción del espacio

Correlación cruzada en un sola dimensión

En esta sección se va a deducir la función de verosimilitud de la suma de dos PDF gaussianas para un caso n -dimensional, pero estrictamente correctas para el caso 1-D. Esto significa, que esta deducción es correcta para mayores dimensiones solo si el alineamiento angular entre los dos datos es conocido.

Como información preeliminar se explicarán los siguientes puntos. Primero, el volumen bajo una gaussiana escalada es igual al factor de escalado

$$\int_{-\infty}^{+\infty} \alpha g(x, \bar{p}, P) dx = \alpha \int_{-\infty}^{+\infty} g(x, \bar{p}, P) dx = \alpha \quad (5.17)$$

Segundo, el volumen obtenido de la multiplicación de dos gaussianas

$$\int_{-\infty}^{+\infty} g(x, \bar{p}, P) g(x, \bar{q}, Q) dx = \int_{-\infty}^{+\infty} \alpha g(x, \bar{r}, R) dx \quad (5.18)$$

es lo siguiente (ver bibliografía para la deducción)

$$\alpha = \frac{1}{(2\pi)^{n/2} \sqrt{|P+Q|}} \exp\left(-\frac{1}{2}(\bar{p}-\bar{q})^T (P+Q)^{-1} (\bar{p}-\bar{q})\right) \quad (5.19)$$

Así pues, dadas dos sumas de gaussianas

$$\begin{aligned} G_0 &= \sum_{i=1}^{k_1} \alpha_i g(x, \bar{p}_i, P_i) \\ G_r &= \sum_{i=1}^{k_2} \beta_i g(x, \bar{q}_i, Q_i) \end{aligned} \quad (5.20)$$

la función de verosimilitud es calculada de la siguiente manera

$$\begin{aligned} \Lambda(x_k) &= G_0(x_k) * G_r(x_k) = \int_{-\infty}^{+\infty} \sum_{i=1}^{k_1} \alpha_i g(u - x_k, \bar{p}_i, P_i) \sum_{j=1}^{k_2} \beta_j g(u, \bar{q}_j, Q_j) du \quad (5.21) \\ &= \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \alpha_i \beta_j \int_{-\infty}^{+\infty} g(u - x_k, \bar{p}_i, P_i) g(u, \bar{q}_j, Q_j) du = \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \alpha_i \beta_j \gamma_{ij}(x_k) \end{aligned}$$

donde la función $\gamma_{ij}(x_k)$ es la correlación cruzada entre las dos gaussianas. de la ecuación 5.19, se deriva que

$$\gamma_{ij}(x_k) = \frac{1}{(2\pi)^{n/2} \sqrt{|P_i + Q_j|}} \exp\left(-\frac{1}{2}(x_k + \bar{p}_i - \bar{q}_j)^T (P_i + Q_j)^{-1} (x_k + \bar{p}_i - \bar{q}_j)\right) \quad (5.22)$$

Correlación cruzada en dos dimensiones

Para el caso de localización 2-D, el estado $x_k = (x_k, y_k, \phi_k)$ varía en ϕ_k y por lo tanto $\Lambda(x_k)$ no puede ser calculada directamente como correlación cruzada. Sin embargo, basta con cambiar la ecuación 5.22 de la siguiente forma

$$\gamma_{ij}(x_k) = \frac{1}{(2\pi)^{n/2} \sqrt{|P'_i + Q_j|}} \exp\left(-\frac{1}{2}(\dot{p}'_i - \bar{q}_j)^T (\dot{P}'_i + Q_j)^{-1} (\dot{p}'_i - \bar{q}_j)\right) \quad (5.23)$$

donde p' y P' son la media y la covarianza del a i -ésima PDF de observación gaussiana transformada al sistema de coordenadas de referencia

$$\begin{aligned} p' &= \begin{pmatrix} x_k \\ y_k \end{pmatrix} + R\bar{p}_i \\ P' &= RP_iR^T \\ R &= \begin{pmatrix} \cos\phi_k & -\sin\phi_k \\ \sin\phi_k & \cos\phi_k \end{pmatrix} \end{aligned}$$

El algoritmo 5.9 indica una simple implementación de la función de verosimilitud, donde $|G|$ indica el número de gaussianas en la suma. Para la suma de dos

de tamaño k_1 y k_2 , este algoritmo tiene una complejidad de $O(k_1k_2)$ y para mayores PDF, pasa a ser muy lento para métodos de estimación como el filtro de partículas. Sin embargo, asumiendo que las gaussianas en cada suma poseen una covarianza y un escalado parecidos, la suma de las verosimilitudes debido a dos gaussianas partuclares son despreciables si la norma de $\|p' - \bar{q}_j\|$ es grande, lo cual lleva a la posibilidad de dos optimizaciones, que dos de ellas se presentan a continuación.

Primero, el espacio de búsqueda(en referencia a la suma de gaussianas) se puede dividir

de tal forma, que para una media p' , solo las gaussianas que satisfagan $\|p' - \bar{q}_j\| < d$ sean incorporadas en el cálculo de la verosimilitud. Esto se puede conseguir aislando los valores de la media \bar{q}_j en una celda gruesa de granularidad $2d$. Alternativamente a este método, se puede conseguir correcto usando los k vecinos más cercanos a p .

La segunda optimización es coger una serie de vecinos cercanos para cada gaussiana en G_0 , y requiere estos mientras el estado x_k esté razonablemente cerca del valor inicial. Esto es útil para la estimación de la verosimilitud máxima, que convergerá incluso si los paquetes de vecinos son actualizados cada n iteraciones.

Alternativa

En el apartado anterior se consideraba la observación y la referencia de PDF's de sumas de gaussianas, pero una alternativa sería considerar cada medida en la observación como independiente. Así pues cada medida tendrá una función de verosimilitud.

$$\Lambda_i(x_k) = \sum_{j=1}^{k_2} \beta_j \gamma_{ij}(x_k) \quad (5.24)$$

donde $\gamma_{ij}(x_k)$ es lo mismo que en la ecuación 5.23. Nótese que el factor de escalado α_i para cada observación no es requerido, dado que cada medida forma una unidad PDF de volumen. Entonces la función de verosimilitud de todo el barrido, será la intersección de las verosimilitudes individuales.

$$\Lambda(x_k) = \prod_{i=1}^{k_1} \Lambda_i(x_k) \quad (5.25)$$

Este método obtiene mejores resultados que los métodos mencionados anteriormente pero también tiene sus propias complicaciones.

Primero, la PDF de referencia es usada para cada iteración en la ecuación 5.24. La correlación entre las verosimilitudes puede ser compensada inflando la incertidumbre de la observación, pero cada observación normalmente solo observa una pequeña porción de la PDF de referencia, lo que dificulta la cantidad de este inflado. Una solución mejor, es quizá, es usar información correlada sin inflar, suponiendo que la ecuación 5.25 va a converger al mejor ajuste de la PDF de referencia y no a un estado falso del entorno.

El segundo problema, es que el barrido de observación no sólo observa partes del barrido de referencia, si no que puede añadir medidas adicionales dada la variación del punto

de vista, oclusión y variación dinámica de la posición de los objetos, que todos estos fenómenos son llamados en su conjunto *outliers*. La presencia de estos *outliers* significa que un valor de estado x_k cercano al valor real puede estar hecho con poca probabilidad por la multiplicación de $\Lambda_i(x_k)$, donde i representa la medida del *outlier*. Esto tiene la particularidad, de que si la medida pose una incertidumbre acotada la probabilidad del estado real puede llegar a ser cero. Un trabajo sobre este problema sería especificar la probabilidad de una falsa detección P_f al valor mínimo de la función de verosimilitud, veáse figura 5.10 En otras palabras, dado un estado particular $x_k = x_0$ es menos que P_f , entonces la medida i es asumida como *outlier* para x_0 .

Este método está en estudio y todavía no ha sido implementado, por lo cual no se tienen medidas experimentales.

Algorithm 4.2: LIKELIHOOD(G_o, G_r, \mathbf{x}_k)

```

 $\Lambda \leftarrow 0$ 
 $\mathbf{t} \leftarrow \begin{bmatrix} x_k \\ y_k \end{bmatrix}$ 
 $\mathbf{R} \leftarrow \begin{bmatrix} \cos \phi_k & -\sin \phi_k \\ \sin \phi_k & \cos \phi_k \end{bmatrix}$ 
for  $i \leftarrow 1$  to  $|G_o|$ 
   $\mathbf{p}' \leftarrow \mathbf{t} + \mathbf{R}\bar{\mathbf{p}}_i$ 
   $\mathbf{P}' \leftarrow \mathbf{R}\mathbf{P}_i\mathbf{R}^T$ 
  for  $j \leftarrow 1$  to  $|G_r|$ 
    do  $\begin{cases} \nu \leftarrow \mathbf{p}' - \bar{\mathbf{q}}_j \\ \mathbf{S} \leftarrow \mathbf{P}' + \mathbf{Q}_j \\ \gamma \leftarrow \frac{1}{2\pi\sqrt{|\mathbf{S}|}} \exp\left(-\frac{1}{2}\nu^T\mathbf{S}^{-1}\nu\right) \\ \Lambda \leftarrow \Lambda + \alpha_i\beta_j\gamma \end{cases}$ 
return ( $\Lambda$ )

```

Figura 5.9: algoritmo para cálculo de la verosimilitud

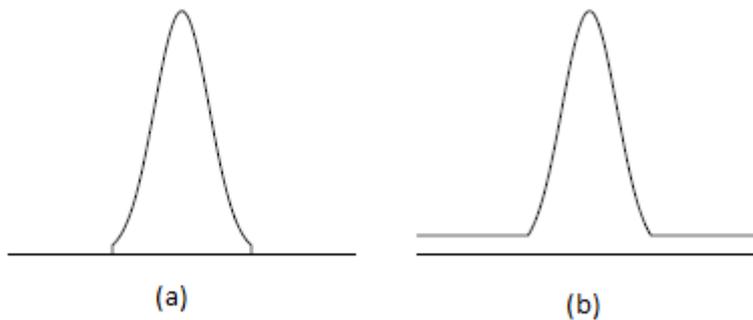


Figura 5.10: Función de verosimilitud con probabilidad de detección de falsa medida. En la figura (a) se observa un a función de verosimilitud nominal, mientras que la (b) tiene probabilidad de falsa medida

Capítulo 6

Resultados experimentales

En este capítulo se va a tratar sobre las pruebas realizadas sobre el algoritmo de la suma de gaussianas (SOG) para el problema de scan-matching. Se realizaron diversas pruebas, con diferentes paquetes de datos, para comprobar si realmente este algoritmo funcionaba bien en todos los casos. Las pruebas realizadas incluían también tiempos de computación para comprobar si los tiempos para lanzar este algoritmo eran lo suficientemente buenos como para ser implementados en un robot real. Se tuvo especial cuidado a la hora de analizar los resultados, teniendo como principal objetivo observar en que puntos acumula más error, y que sucede cuando se intenta cerrar un lazo. Con un paquete de datos se hicieron pruebas variando las distancias y los ángulos para ver que efecto tenía sobre los resultados, y poder estimar cual es la distancia óptima con la que se obtiene un buen resultado, con un tiempo de cómputo óptimo.

6.1. Algoritmo

El algoritmo fue implementado por Tim Bailey en su tesis doctoral, pero esta implementación sólo es probada con un paquete de datos que es tomada de manera muy precisa, y muy seguida, por lo cual el resultado obtenido es muy bueno. En este trabajo se adaptó este algoritmo variando parte de su código para poder trabajar con cualquier tipo de datos, en este caso datos de formato CARMEN. La implementación está hecha en MATLAB, pero tiene una serie de rutinas realizadas en C++, para mejorar los tiempos de cómputo. Este sigue un diagrama de flujo como se muestra en la figura 6.1. Lo primero es transformar los datos a una matriz de dimensiones $m \times n$ donde m es el número de barridos, y n es el número de datos por barrido. Al programa también hay que especificarle el alcance máximo del sensor, y la incertidumbre de la medida. Posteriormente, se procede a inicializar las variables, y tomar el primer barrido como la referencia. Acto seguido se

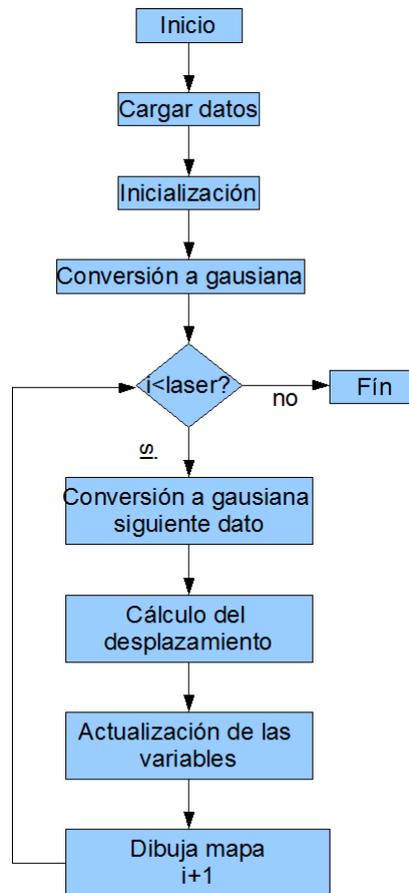


Figura 6.1: Diagrama de flujo del algoritmo de la suma de gaussianas

entra en un bucle recorriendo uno a uno los barridos. Se convierte cada barrido en suma de gaussianas, y se calcula el desplazamiento entre el barrido nuevo y el de referencia. Después se actualizan las variables y se dibuja en el mapa. Y así se repite hasta que se ha recorrido todos los barridos.

6.2. Resultados

En esta sección se van a presentar los distintos resultados obtenidos, y las condiciones en que estos fueron tomados. Se trabajó con 3 paquetes de datos distintos. El primero que es el que propone Bailey en su tesis, que son datos muy buenos, y que apenas se cometen errores, el segundo es de los laboratorios Intel, que vienen dados en formato CARMEN, y es un paquete que el robot da muchas vueltas y los datos fueron tomadas al vuelo, luego es un paquete que acumula muchos errores. Y por último se probó un paquete de datos

obtenidos por el robot Maggie del departamento de ingeniería de sistemas y automática de la universidad Carlos III de Madrid.

6.2.1. Datos de Bailey

Con estos datos es con los que Tim Bailey presenta los resultados de este algoritmo en su tesis. Estos datos fueron tomados con bastante cautela, y por ello los resultados obtenidos con estos datos son los mejores que se han conseguido. El robot estuvo moviéndose por la universidad de Sidney, y en cada barrido obtenía 360 datos. Esto hace que los resultados sean muy buenos, aunque en ciertos casos se comete error, sobre todo en ciertos giros y a la hora de cerrar el bucle. Ahora se presentarán los resultados obtenidos. Se muestran los datos de referencia, el punto dónde se comete el primer error apreciable, un gráfico en el que se vuelve a uno de los puntos iniciales y finalmente el cierre del lazo.

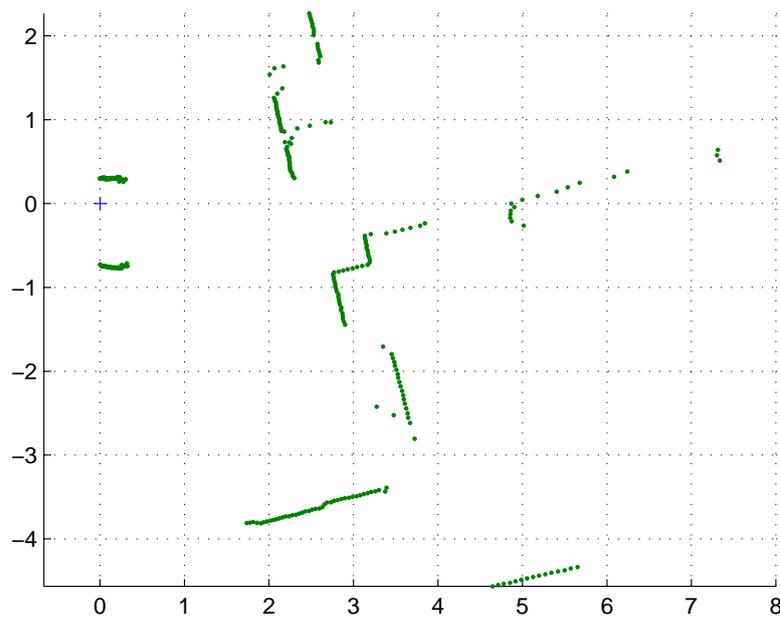


Figura 6.2: Barrido de referencia para los datos de Tim Bailey.

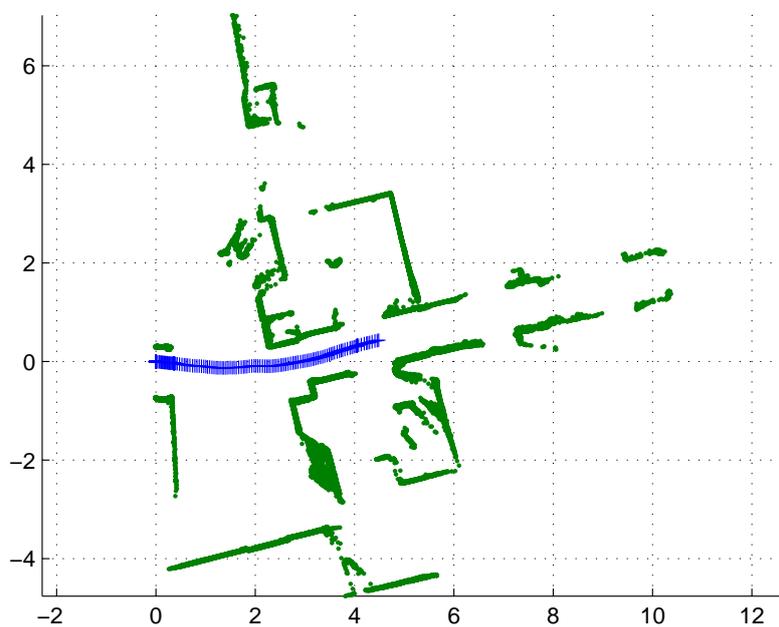


Figura 6.3: Datos a lo largo del pasillo de Tim Bailey

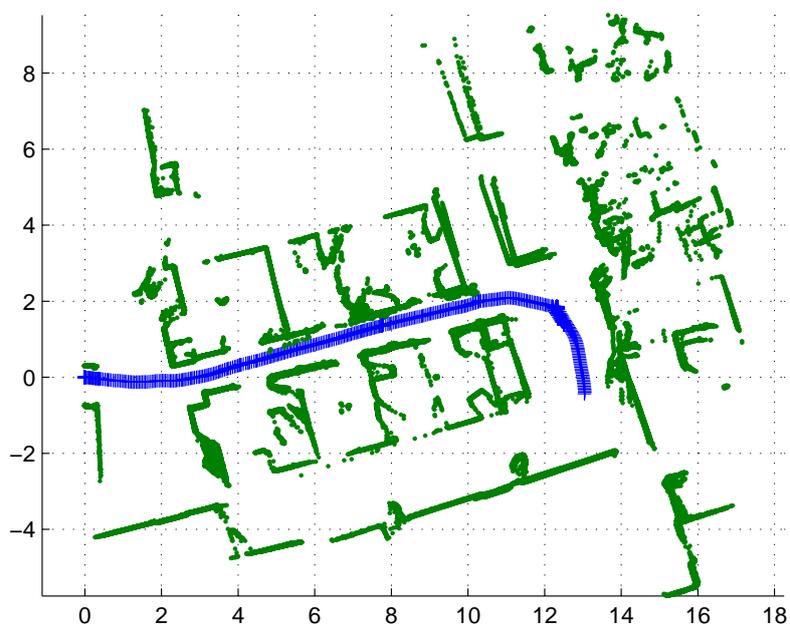


Figura 6.4: Error cometido en el giro de Tim Bailey

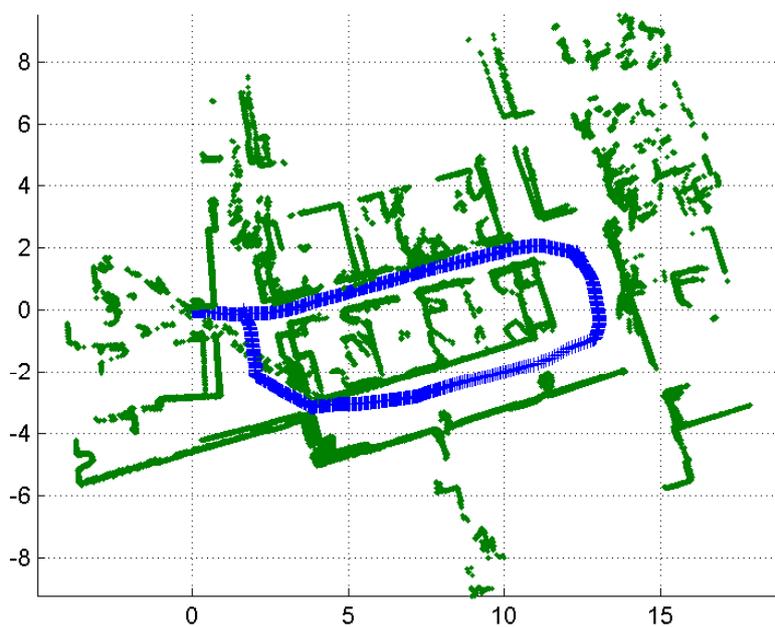


Figura 6.5: Vuelta a uno de los puntos iniciales de Tim Bailey

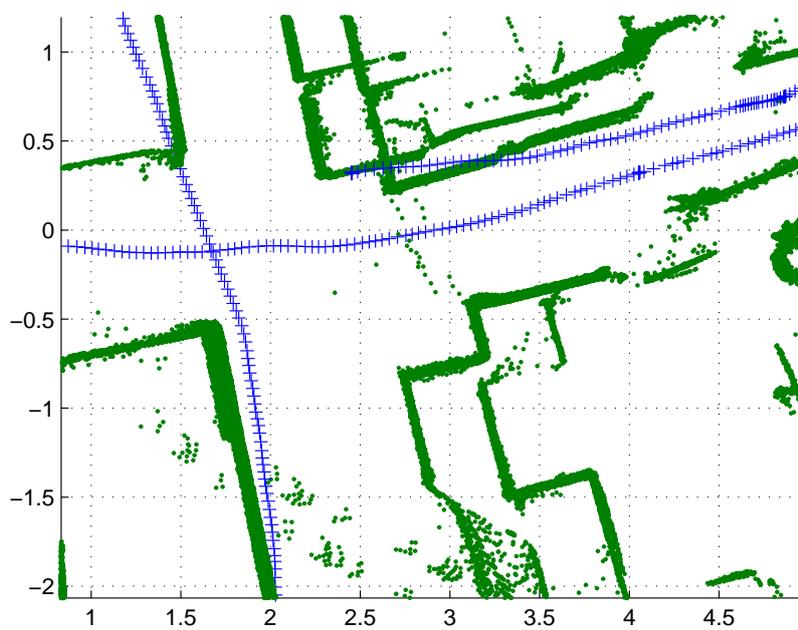


Figura 6.6: Error cometido en el cierre del bucle de Tim Bailey

6.2.2. Datos del laboratorio Intel

Este paquete de datos es un paquete bastante bueno para probar un algoritmo de este estilo, ya que aparte de contener muchísimos datos, el robot recorre una gran distancia cerrando un lazo, luego se puede ver muy bien el error acumulado. De hecho es uno de los distintos paquete estandar para probar este tipo de algoritmos. Con este paquete de datos, se tomaron medidas para un movimiento del robot de 5, 10,20,30,40,50 y 100 cm, viendo que sucedía en cada caso. En este paquete de datos se vió que los datos obtenidos seguían siendo buenos pero no tanto como en los que obtiene Bailey, luego una primera conclusión podría ser que la robustez de este algoritmo dependa de la adquisición de datos. En la figura 6.7 se muestra cómo sería el entorno, y en la figura 6.8 se observa el mapa obtenido al dar una vuelta completa, obtenido con los datos del robot. Se puede observar que según se amplía la distancia entre barridos, el resultado que se obtiene va siendo cada vez peor, y finalmente para un distanciamiento de 1m la desviación es tal que sale una cosa totalmente incongruente. A continuación se muestran los resultados de estos datos al pasarle el algoritmo SOG. Los datos fueron tomados considerando un giro de 0.1 radianes. Como se comentará luego, los mejores resultados fueron los obtenidos para datos separados entre 10 y 20 cm.

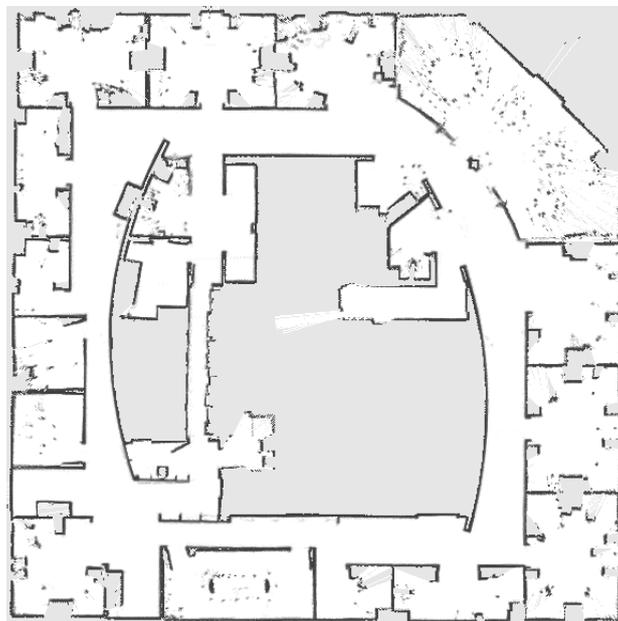


Figura 6.7: Mapa de laboratorio Intel

6.2.3. Resultados de los datos de Maggie

Por último se probó con datos de un robot de esta casa, con datos obtenidos con Maggie. Estos datos fueron adquiridos en la tercera planta del edificio Betancourt de la universidad Carlos III de Madrid. En estos datos solamente se analizará para tener otros resultados más con los que comparar el funcionamiento de este algoritmo. Los resultados se muestran a continuación. Es un entorno complicado, dado que es una habitación con muchas mesas y sillas, luego hay ciertos puntos dónde el alineamiento no es perfecto, y se puede observar que prácticamente se pierde al salir de la habitación. Esto puede ser debido a que los datos de referencia están dentro de la habitación y los barridos de fuera de esta tienen poca o ninguna relación con esta. De todas formas los datos son mostrados a continuación.

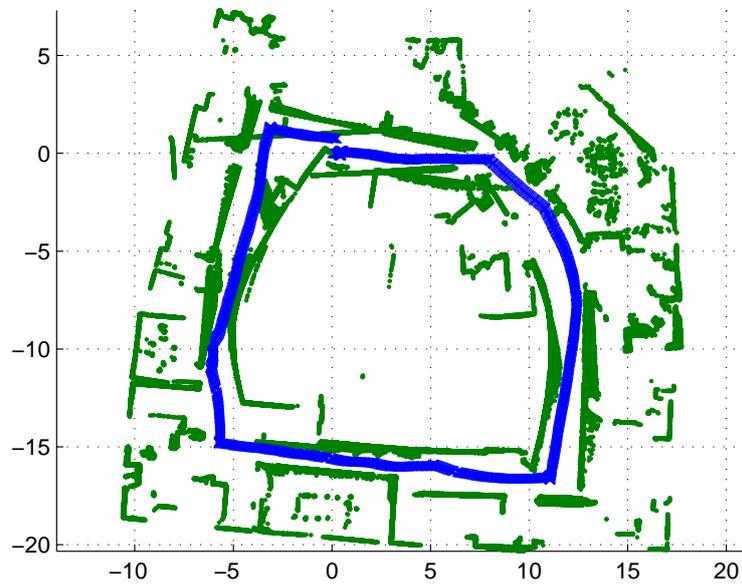


Figura 6.8: Mapa de laboratorio Intel obtenido con los datos aplicando el algoritmo SOG. Se puede apreciar el error cometido con respecto al original

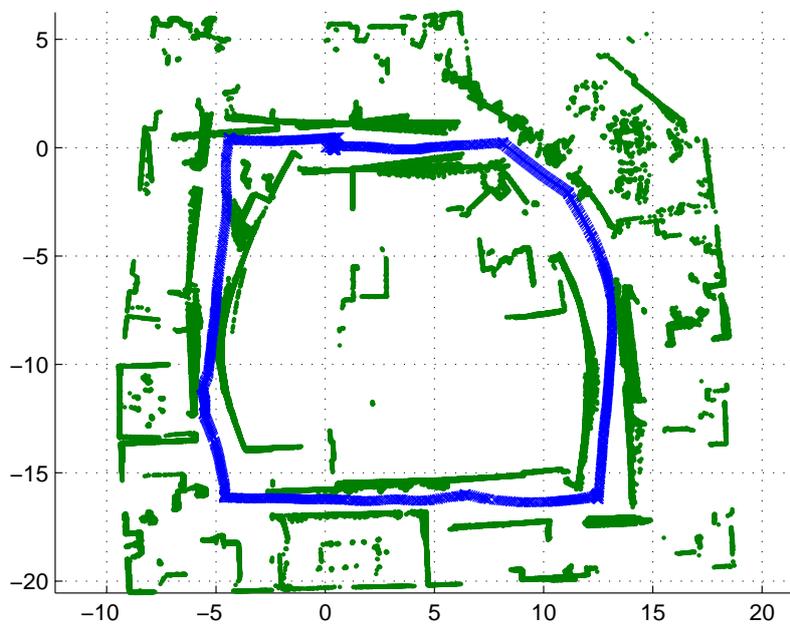


Figura 6.9: Resultado para una vuelta completa con desplazamiento de 5cm

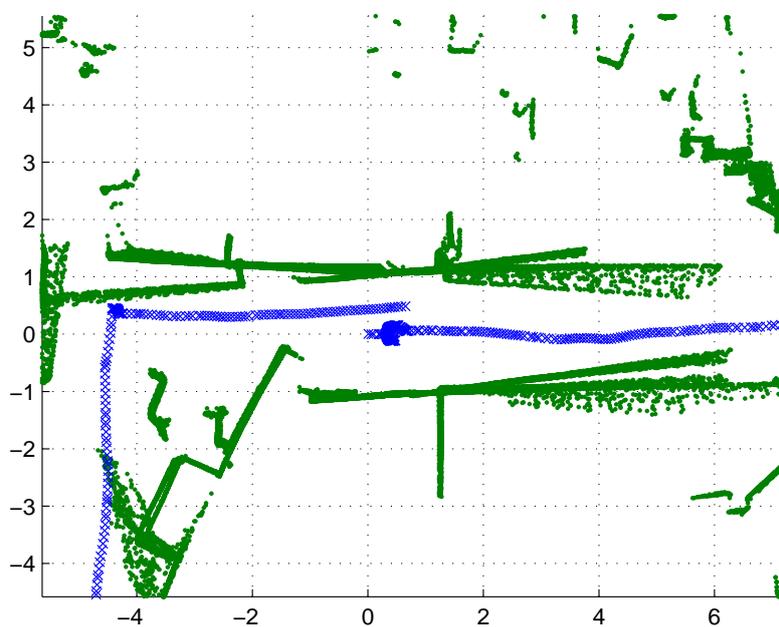


Figura 6.10: Detalle de cierre de lazo para 5cm

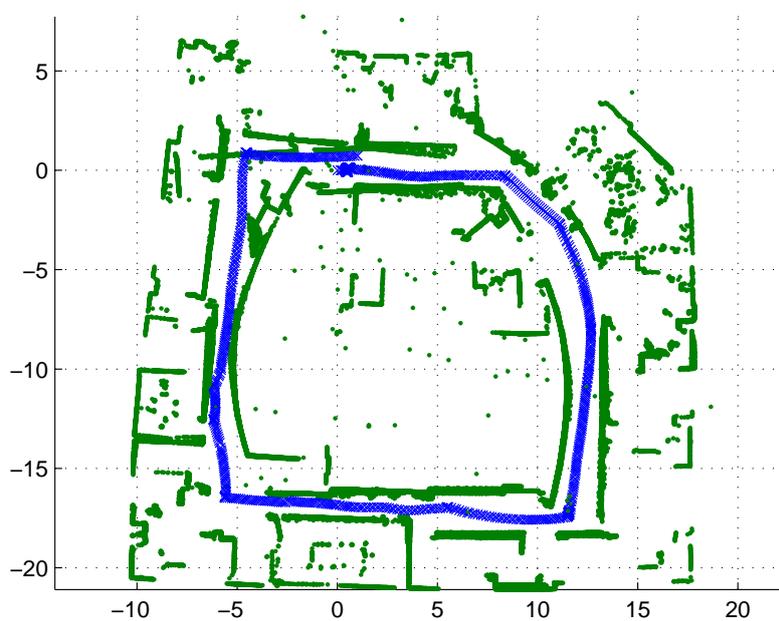


Figura 6.11: Resultado de una vuelta completa con distanciamiento entre medidas de 10 cm

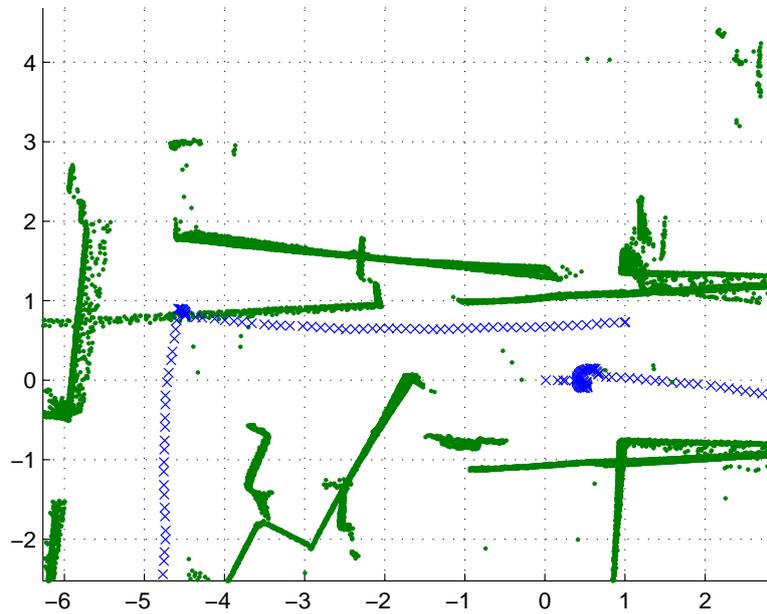


Figura 6.12: Detalle de cierre de lazo para 10 cm. Se puede observar que el error cometido es más grande que para 5 cm

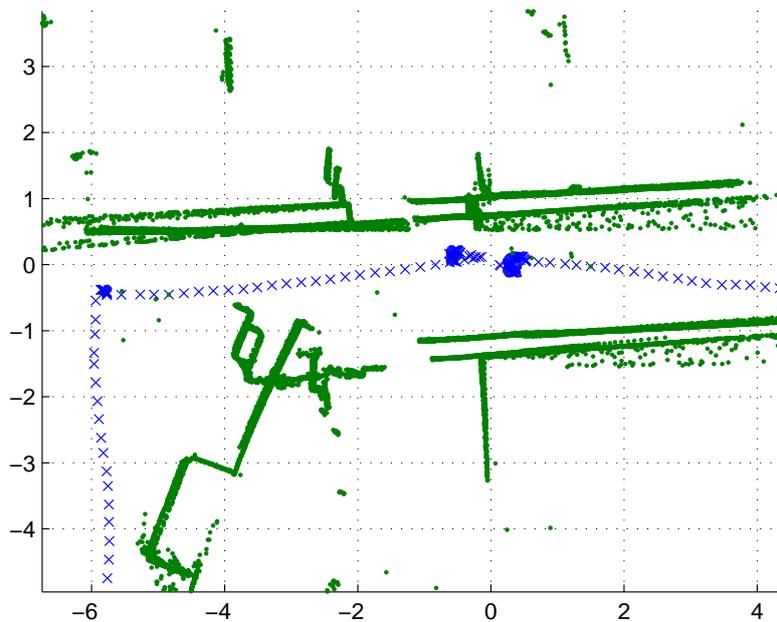


Figura 6.13: Resultado de vuelta completa con distanciamiento entre medidas de 20cm

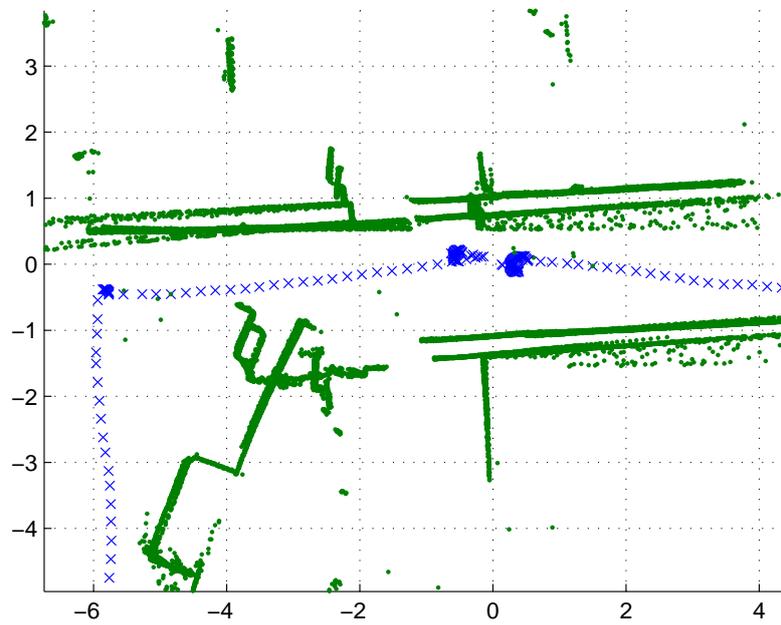


Figura 6.14: Detalle de cierre de lazo para 10 cm. Se puede observar que el error cometido es más grande que para 10 cm

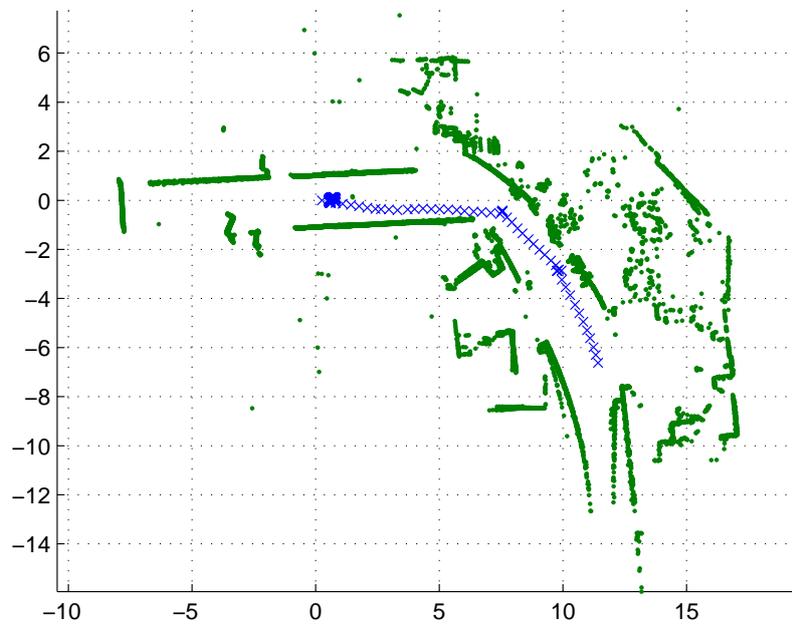


Figura 6.15: Error cometido con un desplazamiento de 30 cm



Figura 6.16: Vuelta completa con un desplazamiento de 30 cm

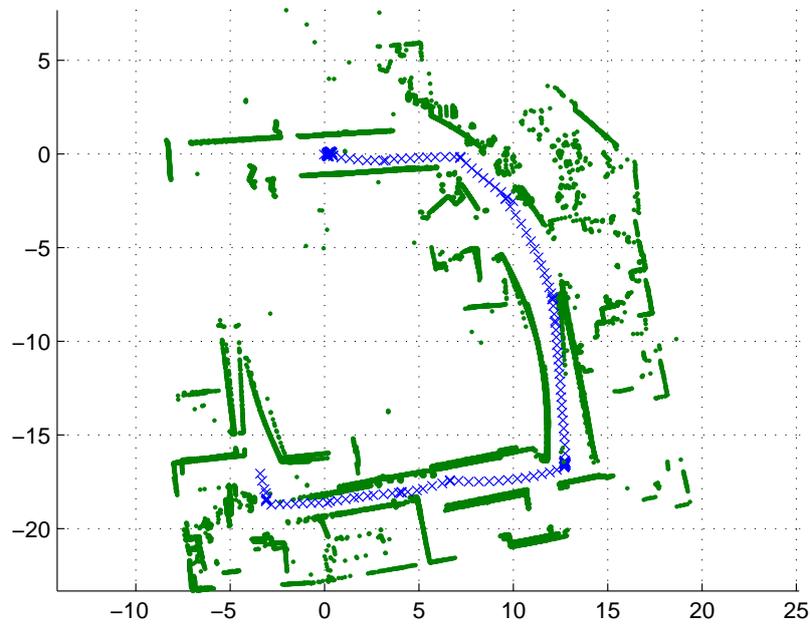


Figura 6.17: Error que se comete con un desplazamiento de 40 cm

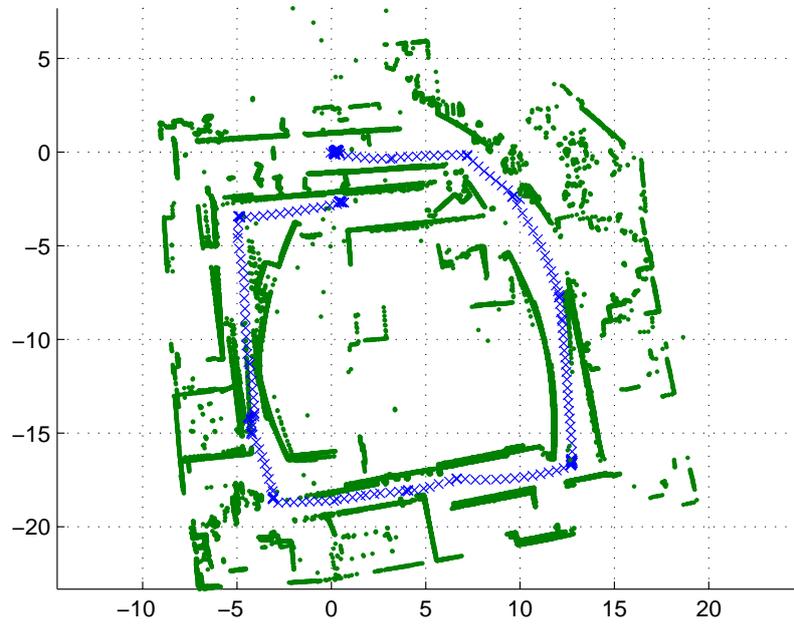


Figura 6.18: Vuelta completa con un desplazamiento de 40 cm

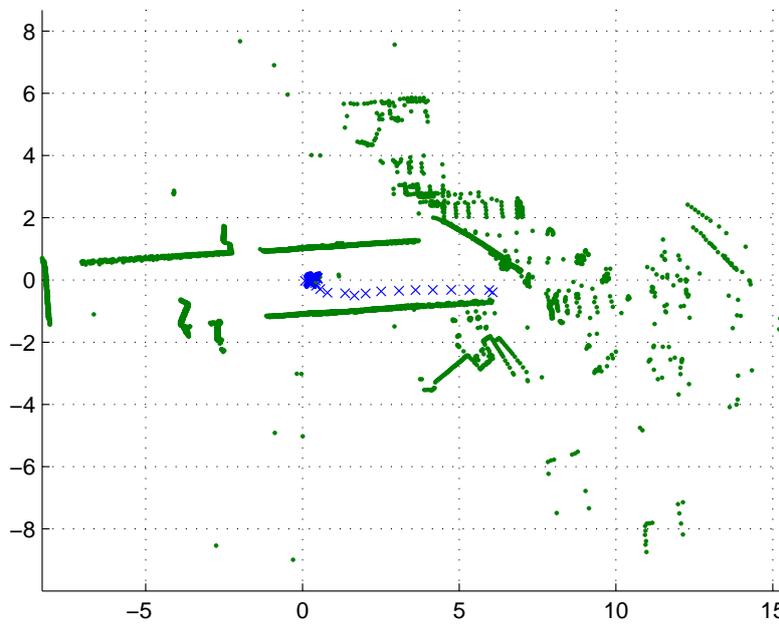


Figura 6.19: Error cometido con un desplazamiento de 50 cm

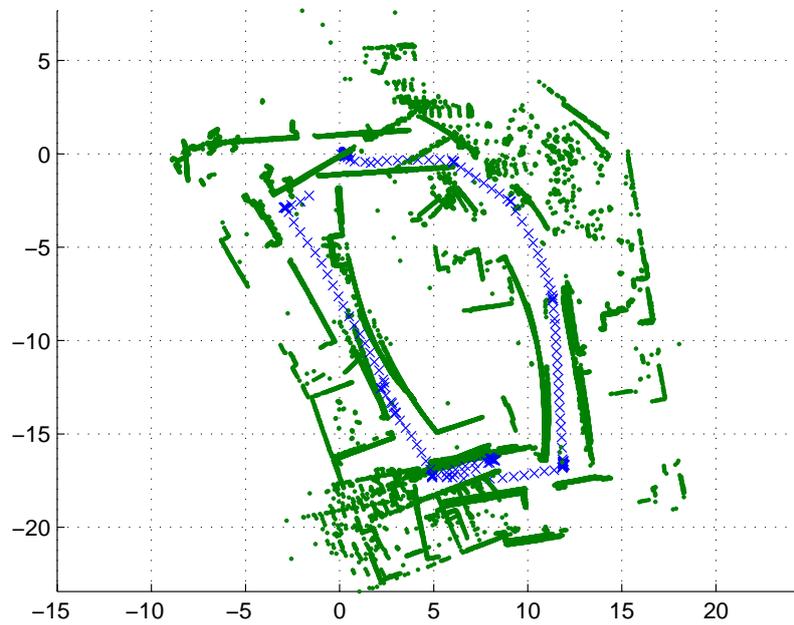


Figura 6.20: Vuelta completa con un desplazamiento de 50 cm. Se puede ver que ya va degenerando demasiado y no se parece a los resultados de los otros desplazamientos

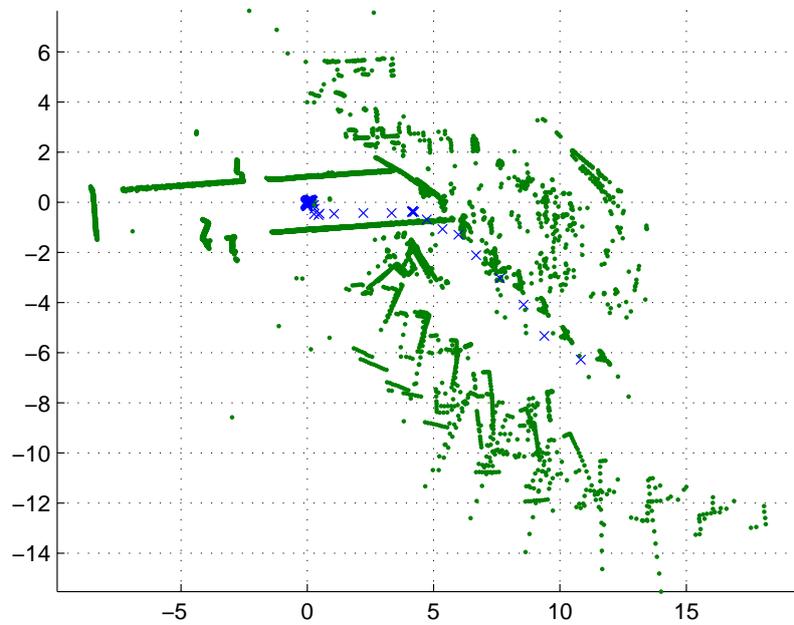


Figura 6.21: Error cometido con un desplazamiento de 100 cm

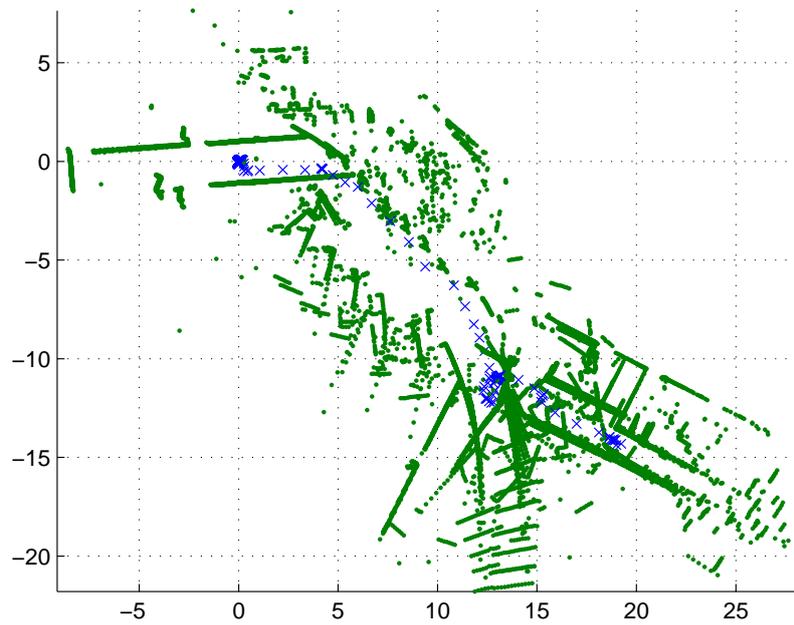


Figura 6.22: Vuelta completa con un desplazamiento de 100 cm. Se puede ver que los resultados no se parecen en nada con los de los otros desplazamientos.

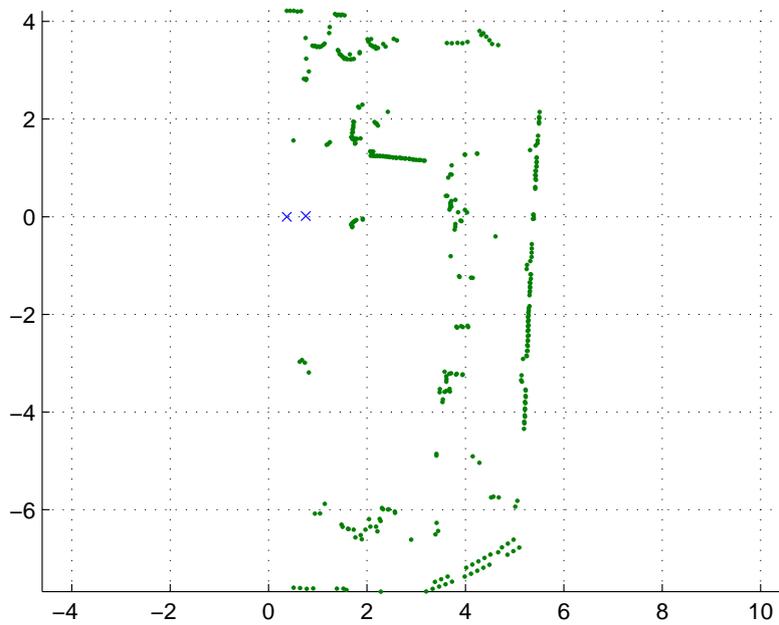


Figura 6.23: Barrido de referencia para Maggie.

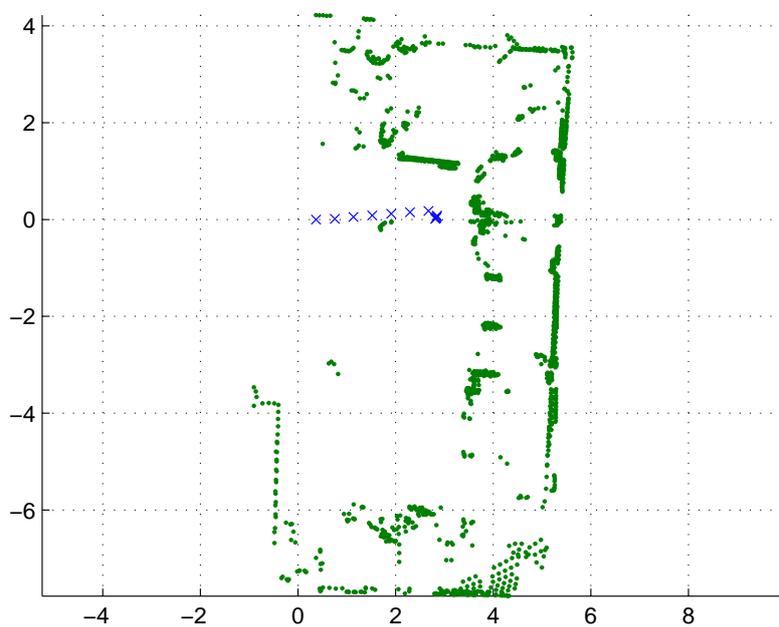


Figura 6.24: Primer giro de Maggie

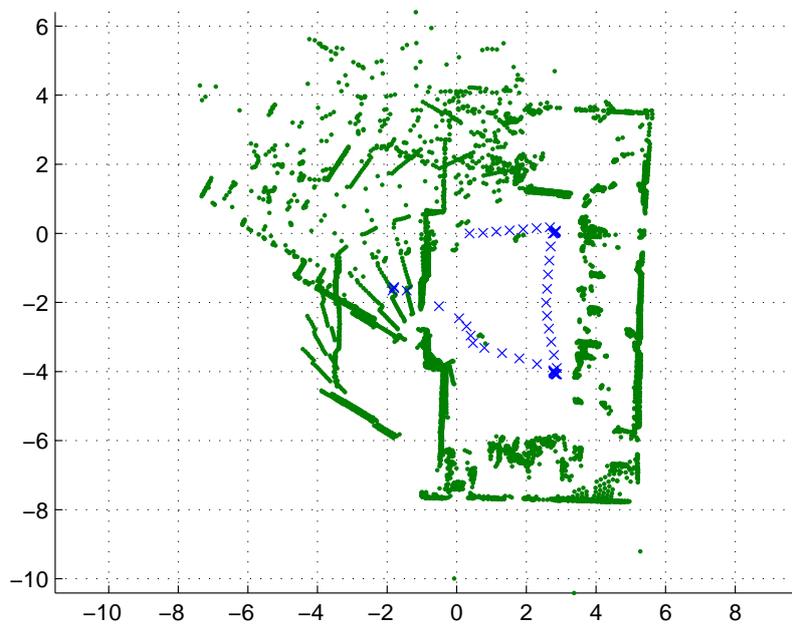


Figura 6.25: Error de Maggie al salir de la habitación

6.3. Resultados de tiempos de computación

En esta sección se comparará los tiempos de computación por iteración para cada uno de los casos mencionados en la sección anterior. Esto es importante para conocer cuánto tiempo tardara este algoritmo en procesar todos los datos, y para ver si puede ser tomado para una implementación sobre el robot. Cabe destacar que cuanto menor sea la distancia entre barridos, mayor será la cantidad de barridos, luego mayor cantidad de datos, por lo tanto se incrementará el tiempo de computación pero se mejorará la estimación de la posición. Será importante buscar un equilibrio entre la precisión y el tiempo de cómputo. En la tabla 6.26 se puede ver los resultados obtenidos. En las columnas están el distanciamiento entre medidas, y en las filas primero el tiempo medio de cómputo, y luego tiempo máximo ambos dos tomados en segundos. Son tiempos bastante pequeños, comparando con otros datos leídos en la bibliografía [6], y con los datos que pude contrastar con mi director en una implementación que está realizando, los datos para este algoritmo son de media 10 veces más rápidos.

	5	10	20	30	40	50	100
Intel	0,1281	0,1242	0,1191	0,1156	0,166	0,1192	0,1222
Maggie	0,1289	0,1132	0,1365	0,1206	0,1257	0,1453	0,153
tmax intel	0,1563	0,1563	0,1563	0,1563	0,1563	0,1875	0,1875
tmax maggie	0,1456	0,1456	0,1456	0,1456	0,1456	0,1763	0,1763

Figura 6.26: Resultados de computación para el algoritmo SOG

Capítulo 7

Conclusiones y trabajo futuro

Como conclusión se puede decir que el comportamiento del algoritmo es bastante bueno. Y que aunque con los datos que proporciona el autor los resultados son mucho mejores, analizando los que se han obtenido en este trabajo, se puede decir que funciona bastante bien. En los datos del mapa de Intel, al cerrar el lazo los mejores se obtienen para resultados para desplazamientos entre 10 y 20 cm. En estos casos los errores de cierre de bucle aunque no se puedan calcular de una manera exacta, de las imágenes se puede observar que estarían entorno a los 2-3m. Con unos resultados de este estilo no se puede hacer una implementación SLAM propiamente dicha, pero ya existen propuestas de lo que es llamado *Scan-SLAM* [4]. Estas técnicas consisten básicamente en primero alinear los datos utilizando algún método de scan-matching y después aplicar un método Slam, como el EKF-Slam explicado en este trabajo. En cuanto a los tiempos por iteración son bastante buenos, comparándolos con los resultados publicados en ciertos papers [6], aunque estos habría que procesar los datos con ordenadores similares para tener datos realistas. Estos datos, y aunque no estén referenciados, comparados con los datos que obtiene Luis Moreno en su investigación con scan-matching, el método de Bailey es del orden de 10 veces más rápido, aunque el resultado es menos correcto. Luego de manera global se puede decir que si se ponderara precisión y coste computacional, este algoritmo obtendría un resultado bastante aceptable.

En cuanto a trabajo futuro, se propone realizar un estado del arte más profundo sobre las distintas técnicas de scan-matching que han ido apareciendo estos últimos años, sobre todo las relacionadas con filtros de partículas, y hacer una implementación de cada una de ellas y compararlas en prestaciones y en tiempos de cómputo. Otra propuesta para continuar con este trabajo sería implementar un algoritmo Scan-SLAM, y ver que resultados se obtiene frente a los métodos basados en asociación de características geométricas.

También otra propuesta sería intentar sacar lo bueno que tiene cada técnica e intentar combinar cada una de las ventajas en un nuevo método de scan-matching. Aunque a día de hoy no se puede hacer un Slam del todo exacto con técnicas de scan-matching, si se sigue investigando quizás algún día se pueda lograr.

Bibliografía

- [1] Tim Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, Australian Centre for Field Robotics, August 2002.
- [2] Aron J. Cooper. A comparison of data association techniques for simultaneous localization and mapping. Master's thesis, Massachusetts Institute Of Technology, June 2005.
- [3] A. Elfes. Sonar-based real-world mapping and navigation. *Robotics and Automation, IEEE Journal of*, 3:249–265, June 1987.
- [4] Eduardo Nebot Juan Nieto, Tim Bailey. Recursive scan-matching slam. *Robotics and Autonomous Systems*, 55:3949, 2007.
- [5] Tim Bailey Juan Nieto and Eduardo Nebot. Scan-slam: Combining ekf-slam and scan correlation. *International Conference on Field and Service Robotics*, 2005.
- [6] Feng Lu and Evangelos Miliotis. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, pages 935–938, 1994.
- [7] Javier Minguéz Luis Montesano and Luis Montano. Probabilistic scan matching for motion estimation in unstructured environments. *Proceedings of the Conference on Intelligent Robots and Systems*, 2005.
- [8] John Leonard Seth Teller Michael Bosse, Paul Newman. An atlas framework for scalable mapping. *IEEE International Conference on Robotics and Automation*, 2003.
- [9] J. D. Tardos J. Neira P. Newman, J. Leonard. Explore and return: Experimental validation of real-time concurrent mapping and localization. *IEEE Int. Conf. Robotics and Automation*, pages 1802–1809, 2002.

-
- [10] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. *Third International Conference on 3D Digital Imaging and Modeling (3DIM 2001)*, pages 145–152, 2001.
- [11] Dieter Fox Sebastian Thrun, Wolfram Burgard. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. *IEEE International Conference on Robotics and Automation*, 1:321–328, April 2000.
- [12] Wolfram Burgard Sebastian Thrun and Dieter Fox. *Probabilistic Robotics*. MIT Press, September 2005.