



UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE CIENCIAS EXACTAS Y NATURALES

DEPARTAMENTO DE COMPUTACIÓN

**SLAM basado en visión estéreo para la
navegación autónoma de robots
hexápodos**

Tesis presentada para optar al título de Doctor de la Universidad de
Buenos Aires en el área Ciencias de la Computación

Thomas Arnim Fischer

Directora de tesis: Marta Mejail

Director Asistente: Dr. Pablo De Cristóforis

Consejero de Estudios: Dr. Julio Jacobo Berlles

Lugar de trabajo: Laboratorio de Robótica y Sistemas Embebidos (LRSE),
Departamento de Computación, Facultad de Ciencias Exactas y Naturales,
Universidad de Buenos Aires

Fecha de defensa: 06 de mayo de 2019

El robot no tenía sentimientos, solo impulsos positrónicos que emulaban esos sentimientos. (Y tal vez los seres humanos no tenían sentimientos, solo impulsos neurólogicos que eran interpretados como sentimientos). – Isaac Asimov, Los robots del amanecer

Índice general

Abstract	1
1 Introducción	3
1.1 Contribuciones	5
1.2 Publicaciones relacionadas	6
1.3 Disposición de los capítulos	7
2 Trabajos relacionados	9
2.1 Robots hexápodos	9
2.2 SLAM probabilístico en tiempo real	11
3 Preliminares	17
3.1 Convenciones de notación	17
3.2 Distribución normal multivariada	17
3.2.1 Normal multivariada particionada	18
3.2.2 Log-verosimilitud	19
3.2.3 Intervalos de confianza	20
3.3 Propagación de la incertidumbre	21
3.4 Transformaciones inciertas	21
3.4.1 Parametrización de transformaciones rígidas	22
3.4.2 Incertidumbre en el espacio euclídeo tridimensional	23
3.5 Métricas de error en SLAM	24
4 Modelos de incertidumbre en VSLAM	27
4.1 Planteamiento del problema	28
4.2 Modelos gráficos	29
4.2.1 Redes Bayesianas	30
4.2.2 Grafos de factores	32
4.3 Estimación de parámetros en SLAM	33
4.3.1 Bundle Adjustment global	34
4.3.2 Seguimiento de pose	38
4.4 Covarianza inicial de un punto	39
4.5 Detección de valores atípicos	41
5 Bundle Adjustment local en tiempo constante	45
5.1 Ventana local de parámetros	46

5.2	Eliminación de parámetros	47
5.2.1	Marginalizando parámetros en ecuaciones de cuadrados mínimos	49
5.2.2	Marginalizar parámetros en problemas malos	51
5.2.3	Actualizando la distribución marginal incrementalmente	53
5.2.4	Prior unario	54
5.3	Seguimiento de pose en tiempo constante	59
6	Sistema de navegación para hexápodos	61
6.1	S-PTAM	62
6.1.1	El mapa	65
6.1.2	Seguimiento de pose	67
6.1.3	Optimización del mapa	68
6.1.4	Corrección de ciclos	69
6.2	Adaptive Motion Gait	70
6.3	Algoritmo de navegación	72
7	Experimentos	73
7.1	Entorno simulado	73
7.2	Plataforma robótica	76
7.2.1	Factibilidad del sistema de SLAM	80
7.2.2	Navegación en terrenos irregulares	84
7.2.3	Navegación autónoma	87
8	Conclusiones	91
8.1	Trabajo futuro	92
	Bibliografía	95

SLAM basado en visión estéreo para la navegación autónoma de robots hexápodos

Para permitir que un robot móvil pueda de navegar y realizar tareas de manera autónoma, previamente se necesitan resolver al menos dos problemas esenciales: Es necesario que exista un control de alto nivel para controlar su desplazamiento, por ejemplo a través de comandos de velocidad, y el robot debe ser capaz de estimar con precisión su pose (posición y orientación) respecto del entorno que lo rodea. Además, para lograr completa autonomía, estos sistemas deben poder ser ejecutados en una computadora a bordo en tiempo real.

En el caso general donde un robot no cuenta con un mapa previo de su entorno ni información externa que le permita conocer su pose, ambas tareas deben ser realizadas en forma simultánea. El problema de localizar a un robot y construir un mapa del entorno simultáneamente se denomina Simultaneous Localization and Mapping (SLAM). Una forma particular de resolverlo es utilizando características visuales extraídas a partir de imágenes capturadas por una o más cámaras montadas sobre el robot. Éstas pueden ser utilizadas para crear un mapa del entorno y localizar al robot dentro del mismo.

El objetivo final de esta tesis es diseñar e implementar un sistema que reúna todas las características necesarias para poder realizar navegación autónoma en robots hexápodos basándose en un sistema de SLAM visual. A pesar de que existen soluciones, por lo general se limitan a espacios de trabajo pequeños y controlados, y no funcionan bien en computadoras con un poder de procesamiento limitado, como las que se encuentran a bordo de los robots móviles. Sobre esta base, se realizan varias mejoras en un algoritmo SLAM del estado del arte para adaptarlos a la tarea particular de la locomoción de hexápodos. La principal contribución de este trabajo se basa en modelar los elementos del problema SLAM basado en minimización como variables aleatorias. Las incertidumbres y la información mutua calculada de esta manera se utilizan para marginalizar de manera eficiente las regiones distantes para Bundle Adjustment minimizando la pérdida de información. Esto evita que el problema escale linealmente con el tamaño del mapa, lo que permite al sistema navegar a largo plazo, incluso en unidades de procesamiento embebidas.

Para validar el enfoque, se realizan una serie de experimentos que muestran un robot hexápodo realizando una navegación autónoma en terreno dificultoso, mientras todas

las tareas se ejecutan en una computadora a bordo y responden en tiempo real. Se miden y analizan diferentes métricas de rendimiento como el costo computacional y el error de localización. Los resultados muestran la viabilidad y la robustez del sistema desarrollado como se propone en esta tesis.

1 Introducción

Los robots con sistemas de locomoción basados en patas articuladas han atraído la atención de la comunidad robótica en los últimos años debido a su capacidad para trabajar en entornos hostiles inaccesibles para robots con ruedas, al mismo tiempo que pueden transportar más carga y operar por más tiempo que los vehículos aéreos no tripulados (UAV). Sus habilidades para realizar movimientos omnidireccionales y atravesar terrenos dificultosos los convierten en una excelente opción para muchas aplicaciones robóticas prácticas, como las misiones de Búsqueda y Rescate Urbanas (USAR) en edificios colapsados, exploración planetaria remota o estudios topográficos de pozos mineros. No obstante requieren de algoritmos de control complejos y mayor consumo energético.

Para navegar en estos escenarios de forma autónoma y sin información previa, es esencial construir una representación espacial del entorno desconocido, el cuál es referido como *mapa*. Además, la pose relativa (posición y orientación) del robot respecto del mapa construido también debe ser estimada. En la comunidad robótica, la solución simultánea de estos dos problemas mutuamente dependientes se conoce como Localización y Mapeo Simultáneos (SLAM) [1]. Un método SLAM eficiente y confiable es un elemento central de un sistema de navegación autónomo para robots móviles, y proporciona tanto la información de pose como el modelo del entorno necesario para la planificación y el control de movimiento eficiente de un robot autónomo.

Para lograr un alto grado de consistencia global, se deben utilizar sensores exteroceptivos para observar el entorno circundante. En particular, los sensores visuales como cámaras y sensores de rango de luz estructurados (SLRS) o escáneres láser suelen ser las opciones más populares para SLAM. Por lo general, las cámaras son más baratas, más pequeñas y más livianas, y aún brindan datos 3D del entorno en alta resolución en tiempo real y requieren un menor consumo de energía que las alternativas. Además, a diferencia de sus contrapartes, no se limitan a ambientes interiores. Sin embargo, los datos recibidos por las cámaras requieren un procesamiento más complejo que la información recibida por otro tipo de sensores, por lo que requieren algoritmos más elaborados y producen una mayor carga computacional.

Soluciones de SLAM visual robustas y precisas [2, 3, 4, 5] han existido por algún tiempo. Algunas de estas técnicas incluso se han implementado con éxito en muchos tipos de robots, por ejemplo, con ruedas [6], voladores [7], con patas articuladas [8], o incluso vehículos submarinos [9]. No obstante, los robots con patas constituyen un desafío particular para las técnicas de localización basadas en la visión porque el mo-

vimiento de la cámara entre fotogramas consecutivos puede ser impredecible debido a las irregularidades en la marcha, el deslizamiento y el terreno. En este contexto, la precisión de un sistema Visual SLAM en tiempo real depende en gran medida de la calidad de un modelo de predicción de movimiento entre cuadros consecutivos. Se usa con frecuencia un modelo de movimiento suave (por ejemplo, disminución de velocidad lineal), pero generalmente se ve trastornado por los movimientos repentinos propios de una plataforma robótica caminante. Las alternativas populares incluyen la incorporación de sensores de movimiento simples y fiables a corto plazo, como los codificadores de actuador o las unidades de medición inercial (IMU) para superar este problema.

En los últimos años, la tendencia ha sido fusionar las técnicas de Visual SLAM con sensores inerciales [10, 11], motivados principalmente por el apogeo de las aplicaciones orientadas a realidad aumentada y virtual [12], así como a los sistemas de localización autónomos diseñados para vehículos aéreos no tripulados (UAV) [13, 14]. En el contexto de las restricciones computacionales impuestas en ambos casos por la necesidad de ejecución en tiempo real en sistemas integrados y los rápidos movimientos repentinos inherentes a los dispositivos de mano alzada y UAVs, las mediciones inerciales pueden proporcionar predicciones confiables a corto plazo de estos movimientos. Estas predicciones se pueden usar como soluciones iniciales del sistema visual para acelerar y mejorar los algoritmos de correspondencia de mapas. Dado que la ejecución de SLAM visual en robots con patas que se mueven en terrenos irregulares sufre de problemas similares, parece razonable que el mismo enfoque también pueda ser utilizado para superarlos.

Sin embargo, las soluciones de localización basadas en visión aún requieren de considerables recursos de cómputo, lo cuál se convierte en un problema al utilizar a robots de baja capacidad de carga, como vehículos autónomos pequeños, donde solo se puede emplear hardware con capacidades de procesamiento limitadas. Las operaciones típicas de Visual SLAM, como la asociación de datos y optimización, dependen en gran medida del tamaño del mapa considerado. Dado que el mapa puede potencialmente crecer hasta miles de marcas de referencia y fotogramas clave, estas tareas podrían no llegar a ejecutarse en tiempo real en una computadora a bordo si se considera todo el mapa. Por lo tanto, se requiere un método eficiente para determinar qué parte del mapa es relevante en cada paso sin sacrificar demasiada precisión y robustez.

La noción de co-visibilidad [15] es una herramienta útil para determinar los fotogramas clave y los puntos del mapa que están estrechamente relacionados con una posición de cámara determinada en un tiempo constante. Se dice que un par de fotogramas clave son co-visibles hasta el grado n si observan al menos n puntos del mapa en común. Sin embargo, sigue sin quedar claro qué fotogramas clave y marcas de referencia del vecindario deben seleccionarse para lograr los mejores resultados, o cómo descartar otros elementos sin perder demasiada información. Al modelar el problema de SLAM como un problema de inferencia probabilística utilizando representaciones gráficas típicas [16] como redes de creencias o grafos de factores,

el problema puede abordarse en términos de minimizar la pérdida de información probabilística.

Además, al estimar cantidades del mundo real utilizando datos de sensores ruidosos, la estimación de incertidumbre de esas cantidades se convierte en un requisito esencial para realizar tareas autónomas de manera segura y confiable. Al aprovechar los modelos probabilísticos antes mencionados, las regiones de incertidumbre y confianza pueden estimarse adecuadamente. Sin embargo, se debe tener cuidado en cuanto a cómo se calculan estas incertidumbres cuando se utilizan heurísticas locales para aproximar la solución global del problema, como es el caso de usar una ventana para realizar una optimización de tiempo constante.

En esta tesis, se pretende desarrollar un sistema de navegación autónoma para una plataforma de robot hexápodo, basada en una versión mejorada del sistema de SLAM visual S-PTAM [3]. Se ha desarrollado un marco probabilístico para permitir la localización y el mapeo de tiempo constante y probabilísticamente coherente, lo que permite que el sistema se ejecute de forma autónoma y responda en tiempo real en plataformas integradas con bajos recursos computacionales. La localización es fusionada con la información proporcionada por una IMU para compensar los movimientos perturbadores inherentes a los sistemas de locomoción de robots con patas que navegan en terrenos dificultosos. Para finalizar, se realizan una serie de experimentos en una plataforma robótica real que muestran la viabilidad del sistema propuesto.

1.1. Contribuciones

La contribución general de esta tesis es el diseño, la implementación y la validación experimental de un sistema de navegación para un robot hexápodo. Se pueden destacar las siguientes contribuciones a subsistemas específicos:

- El autor contribuyó en gran medida al desarrollo del sistema de SLAM visual S-PTAM [3], específicamente diseñado para la ejecución en tiempo real en sistemas integrados con capacidad de procesamiento limitada. Una contribución importante es la introducción de la llamada estructura de mapa de co-visibilidad que permite una marginalización eficiente de partes relevantes del mapa.
- Se propone y valida un modelo de incertidumbre probabilística de las poses y posiciones de marcas de referencia publicadas por S-PTAM. El modelo puede potencialmente generalizarse a cualquier algoritmo SLAM basado en optimización.
- Se propone una política basada en la co-visibilidad y probabilidad para la marginalización eficiente del mapa que permite ejecutar *Bundle Adjustment* en tiempo real, minimizando la pérdida de información con respecto a la solución canónica

Las herramientas desarrolladas se implementaron e incorporaron en un sistema de navegación completa, que se implementa en una plataforma robótica hexápoda real. Una serie de experimentos muestran que el robot navega de manera autónoma en un terreno accidentado mientras ejecuta el sistema de navegación en tiempo real en una computadora integrada.

1.2. Publicaciones relacionadas

A continuación se describen brevemente las publicaciones realizadas por el tesista durante el transcurso de la tesis, así como los aportes realizados por el mismo a cada uno de ellas.

- G. I. Castro, M. A. Nitsche, T. Pire, T. Fischer and P. De Cristóforis: **Efficient on-board Stereo SLAM through constrained-covisibility strategies**. In Robotics and Autonomous Systems (RAS). March 2018. **Aporte:** Contribución al desarrollo e implementación de un sistema de selección de ventanas en tiempo constante.
- M. A. Nitsche, G. I. Castro, T. Pire, T. Fischer and P. De Cristóforis: **Constrained-covisibility marginalization for efficient on-board stereo SLAM**. In European Conference on Mobile Robots (ECMR). Paris, France, September 2017. **Aporte:** Contribución al desarrollo e implementación de un sistema de selección de ventanas en tiempo constante.
- F. Pessacg, T. Fischer, M. A. Nitsche: **External localization system for mobile robots using multiple cameras**. IX Jornadas Argentinas de Robótica (JAR). Córdoba, Argentina, Agosto 2017. **Aporte:** Contribución a la implementación de un sistema de localización externo basado en cámaras y patrones gráficos.
- P. De Cristóforis, T. Fischer, M. Nitsche: **ICP covariance computation for differential robot localization using laser and wheel odometry**. IX Jornadas Argentinas de Robótica (JAR). Córdoba, Argentina, Agosto 2017. **Aporte:** Implementación de un método de estimación de covarianzas para algoritmos de SLAM basados en mediciones láser y optimización ICP.
- T. Pire, T. Fischer, G. I. Castro, J. Civera, P. De Cristóforis and J. Jacobo Berles: **S-PTAM: Stereo Parallel Tracking and Mapping**. In Robotics and Autonomous Systems (RAS). September 2016. **Aporte:** Contribución al desarrollo e implementación de un sistema de SLAM basado en cámaras estereo.
- T. Fischer, T. Pire, P. Čížek, P. De Cristóforis and J. Faigl. **Stereo Vision-based Localization for Hexapod Walking Robots Operating in Rough Terrains**. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Daejeon, Korea, October 2016. **Aporte:** Desarrollo de

un sistema de SLAM visual-inercial adaptado para la navegación de robots hexápodos en terrenos accidentados. Ejecución de experimentos.

- T. Pire, T. Fischer, J. Civera, P. De Cristóforis and J. Jacobo Berles: **Stereo Parallel Tracking and Mapping for Robot localization**. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Hamburg, Germany, September 2015. **Aporte:** Contribución al desarrollo e implementación de un sistema de SLAM basado en cámaras estéreo.
- T. Pire, T. Fischer and J. Faigl: **Impact assessment of image feature extractors on the performance of SLAM systems**. In Proceedings of the Student Conference on Planning in Artificial Intelligence and Robotics. Organized by Czech Technical University, Faculty of Electrical Engineering. Písek, Czech Republic, September 2015. **Aporte:** Contribución a la implementación del sistema de evaluación y la ejecución de experimentos.

1.3. Disposición de los capítulos

En el capítulo 2 se revisa la bibliografía existente sobre los temas de navegación con hexápodos y SLAM probabilístico. Se discuten diferentes enfoques y se destacan las principales diferencias con el trabajo actual. En el capítulo 3 se explican algunos conceptos y convenciones esenciales utilizados a lo largo de la tesis. El capítulo 4 propone un marco probabilístico para modelar el problema de localización y una solución canónica probabilísticamente consistente. Sin embargo, esta solución no es práctica porque la complejidad computacional crece con el tamaño del entorno explorado. El capítulo 5 propone una heurística que permite realizar la optimización del mapa con una complejidad constante al mismo tiempo que se aproxima la solución canónica de manera probabilísticamente consistente. En el capítulo 6 se describen detalladamente los diferentes componentes del sistema de navegación autónoma desarrollada. En el capítulo 7, se evalúa el rendimiento del método propuesto utilizando simulaciones y experimentos en una plataforma robótica real. En el capítulo 8 se extraen algunas conclusiones sobre los resultados del capítulo anterior, y se proponen trabajos futuros para continuar con la línea de trabajo presentada en esta Tesis.

2 Trabajos relacionados

2.1. Robots hexápodos

Los primeros ejemplos de robots hexápodos se remontan a la década de los 80 [17], utilizando patrones de andar simples de circuito abierto, como el trípode, inspirados en la observación de los patrones de andar de insectos comunes. Poco después, Kaneko et al. [18, 19] presentó algoritmos de control de lazo cerrado mediante el uso de retroalimentación de sensores de fuerza en la punta de cada pie, así como un robot hexápodo que presentaba patas articuladas con 3 grados de libertad (DoF, por sus siglas en inglés). Trabajos posteriores se centraron principalmente en generadores de marcha alternativos, por ejemplo usando redes neuronales artificiales [20] o algoritmos genéticos [21], y control basado en la orientación y la posición mediante cinemática inversa [22, 23].

Los patrones de movimiento tienen una influencia importante en el rendimiento general de un robot caminante, afectando características como la capacidad de desplazamiento, el consumo de energía y la estabilidad. Por otra parte, los patrones tradicionales mencionados anteriormente (por ejemplo, la marcha de trípode) generalmente no pueden atravesar terrenos difíciles. Esto ha llevado al desarrollo de una variedad de métodos que utilizan diferentes niveles de percepción del terreno para controlar el progreso y la estabilidad del robot. Se puede hacer una distinción importante entre los enfoques proactivos y reactivos. El primero requiere una representación densa del terreno transitable para calcular las colocaciones y movimientos óptimos del pie para cada una de las piernas, que tiende a ser computacionalmente costosa y no tiene en cuenta las ubicaciones inestables. Por otro lado, los métodos reactivos detectan colisiones en el suelo utilizando sensores propioceptivos, pero carecen de evasión de obstáculos y, por lo tanto, pueden considerarse una extensión de los patrones de onda a terrenos irregulares e inestables. También existen estrategias híbridas, que intentan compensar mutuamente las desventajas de ambos enfoques.

Sobre este tema, Mrva y Faigl [24] proponen un enfoque reactivo para ajustar los parámetros de la marcha mediante la detección táctil utilizando solo información de posición de los servos. Esto minimiza los recursos computacionales dedicados a la selección de la marcha a la vez que se mantiene el cuerpo del hexápodo estabilizado.

Los primeros intentos de localización con robots hexápodos [25] hacían uso de la odometría con patas, unidades inerciales y sensores GPS para determinar la pose del robot. Las mediciones típicamente se integraban utilizando filtros EKF. Sin embargo,

la odometría con patas a menudo devuelve estimaciones de pose erróneas debido a la presencia de errores no sistemáticos como superficies deslizantes e imperfecciones mecánicas.

Casi al mismo tiempo, el primer sistema de SLAM visual en tiempo real [26] se estaban convirtiendo en una realidad. Este sistema se basó en EKF y utiliza una cámara monocular para extraer puntos salientes del entorno. Los primeros esfuerzos para desplegarlo en robots caminantes se realizaron en plataformas humanoides [27], aunque se mejoraron al alimentando el filtro con las entradas de control del robot, así como la integración de mediciones de un giroscopio. Schmidt y Kasiński [8] fueron los primeros en implementar un sistema de SLAM visual sobre una plataforma hexápoda [28], utilizando una versión más madura del mismo sistema de SLAM visual [5] y también incorporando las entradas de control y mediciones de giroscopio para ayudar a las predicciones del filtro. La configuración fue capaz de lograr la localización a una tasa de 2 estimaciones por segundo.

Al mismo tiempo, Chilian y Hirschmuller [29] presentan un enfoque basado en visión estéreo para la navegación con hexápodos. Aunque no realiza SLAM, utilizan odometría visual para estimar la pose del robot, que a su vez se utiliza para crear un mapa de altura del entorno mediante la extracción de información de profundidad de una imagen de disparidad. La capacidad de traversabilidad de cada celda del mapa se calcula analizando la pendiente, la rugosidad y el tamaño del paso, y un planificador D^* lo utiliza para realizar una navegación autónoma en distancias cortas. Sobre la base de este trabajo, Stelzer et al. [30] incorporan la odometría de las piernas y mediciones inerciales, que se fusionan con la odometría visual mediante un filtro de información indirecto para lograr una estimación de pose más robusta. La capacidad de traversabilidad del terreno se estima utilizando un algoritmo de ajuste de plano, y también se utiliza para elegir el modo de caminar más adecuado. Los autores enfatizan específicamente la necesidad de calcular los valores de incertidumbre para las estimaciones de pose y mapa para mejorar la solidez de la solución.

Belter y Skrzypczynski [31, 32] utilizan el novedoso método de SLAM visual basado en *Bundle Adjustment* PTAM [33, 34] para la localización argumentando que los enfoques basados en filtros no son tan precisos y no escalan bien en la navegación a largo plazo. Además de la cámara utilizada para la localización, se monta un escáner láser para recuperar un mapa de elevación 2D utilizado en la planificación del movimiento de las piernas para pisar puntos de apoyo precisos. Luego de una caracterización del sistema PTAM, se agrega un sistema AHRS de alta gama para predecir la orientación del robot, ya que la orientación proporcionada por el método SLAM no es suficiente para mapear con precisión la estructura del terreno. La configuración se prueba en una maqueta de terreno irregular con el robot hexápodo Messor [35]. Sin embargo, el sistema sigue utilizando una sola cámara y, por lo tanto, se basa en la calidad de una calibración inicial entre dos poses, similar a la de una cámara estéreo, para recuperar la escala del entorno. Tampoco queda claro si el sistema es capaz de procesar datos en tiempo real.

Más recientemente, Bjelonic et al. [36] utilizan el sistema de odometría visual-inercial OKVIS en su robot hexápodo Messor para realizar una navegación autónoma en un terreno irregular. Su principal contribución proviene de un algoritmo híbrido (proactivo y reactivo) de percepción del terreno basado en información visual e inercial. La caracterización se utiliza para adaptar los parámetros típicos del patrón de onda (rigidez de la articulación y altura y frecuencia de zancada) al terreno específico, evitando así la planificación computacionalmente costosa de la trayectoria para las puntas de los pies individuales. Sin embargo, esto todavía requiere considerables recursos computacionales.

Según el conocimiento de los autores, todavía no existe un sistema SLAM basado en imágenes capaz de ejecutarse en tiempo real que se haya implementado con éxito en robots hexápodos. La mayoría de los esfuerzos realizan solo odometría visual o no alcanzan una tasa de cuadros razonable y no escalan bien a medida que aumenta el área explorada. También hay poca o ninguna mención de las técnicas de estimación de incertidumbre, que son fundamentales para realizar tareas autónomas de manera robusta. Esto motiva una exploración más profunda de los sistemas de localización visual de vanguardia que cumplen con los requisitos antes mencionados.

2.2. SLAM probabilístico en tiempo real

SLAM ha sido un tema activo de investigación en robótica durante las últimas décadas, y el trabajo reciente se ha centrado cada vez más en el rendimiento en tiempo real de las aplicaciones. Debido al tamaño potencialmente ilimitado del mapa que crece con respecto al tiempo y el espacio explorado, realizar una optimización ideal sobre todas las variables consideradas se vuelve rápidamente intratable. Por lo tanto, se deben definir heurísticas para seleccionar solo las partes del mapa relevantes para la optimización, mientras se mantiene el resultado lo más globalmente consistente posible. Estos también se conocen comúnmente como algoritmos de complejidad en tiempo constante.

A fines de los años 80, se comenzaban a introducir métodos probabilísticos en robótica y se realizaron trabajos fundamentales [37, 38] para establecer una base estadística para razonar sobre la geometría del espacio en términos probabilísticos. En este contexto, las formulaciones iniciales de SLAM [39, 40, 41] se diseñaron para hacer frente a la incertidumbre inherente en las mediciones de sensores en el mundo real, y, por lo tanto, se definieron como problemas probabilísticos de optimización. Estos enfoques iniciales se basaron en filtros probabilísticos, principalmente el Filtro de Kalman Extendido (EKF) donde el estado (también denominado mapa) está formado por la pose actual del robot y las posiciones de todas las marcas de referencia detectadas. En cada paso, el estado se actualiza mediante la estimación de máxima verosimilitud de un modelo probabilístico del estado incorporando las últimas observaciones. La principal limitación de estos enfoques es la complejidad computacional relacionada con la actualización de la matriz de varianza-covarianza del estado. Dado que las

posiciones de marcas de referencia están altamente correlacionadas estas matrices son potencialmente muy densas, y por lo tanto cuadráticas en el número de marcas de referencia detectadas. Otro inconveniente es que las funciones de observación y transición no lineales no se pueden volver a linealizar una vez incorporadas en el filtro.

Smith y Cheeseman [37, 39] también fueron los primeros en razonar acerca de la incertidumbre de las mediciones que modelan las relaciones espaciales entre objetos en el espacio euclidiano. Estas incertidumbres surgen principalmente debido a las limitaciones físicas de los sensores y las aproximaciones numéricas discretas. Específicamente, los modelos de incertidumbre probabilística se aplican al problema de la navegación de robots en 2D y 3D utilizando estimaciones de marcas ruidosas a partir de mediciones de sensores. El objetivo es poder razonar sobre la base de información inexacta sobre las relaciones espaciales entre los objetos, a fin de reducir la incertidumbre de la ubicación (tanto en la posición como en la orientación) en un grado suficiente para lograr tareas particulares. Por un lado, se presenta el concepto de un mapa relacional de poses de robot y otros objetos, como marcas de referencia, conectados por una red de relaciones espaciales (inciertas). Por otro lado, dos operaciones básicas de estimación se definen en esta estructura similar a un grafo que permiten la estimación de la relación entre dos marcos de coordenadas, dada la transformación relativa incierta que las vincula. La primera operación, llamada composición, permite que una cadena de transformaciones se contraiga (recursivamente) en una sola. La transformación final compuesta debe mostrar una incertidumbre mayor que la de cualquiera de sus componentes. La segunda operación, llamada fusión, combina información de dos o más transformaciones entre el mismo par de objetos para producir una única resultante, con una incertidumbre más baja que cualquiera de las originales.

Casi una década después, FastSLAM [42, 43] introduce los filtros de partículas como herramienta para modelar procesos no lineales y distribuciones no gaussianas en el contexto de SLAM, específicamente utilizando un filtro de partículas de Rao-Blackwell. También fue pionero en la idea de abordar el problema SLAM desde un punto de vista bayesiano, destacando las independencias condicionales entre las variables. Esto permite condicionar las densidades de marcas del ambiente sobre toda la trayectoria, lo que permite tratar las posiciones de las marcas como variables independientes y estimarlas por separado con filtros EKF independientes. Al descomponer el problema en un problema de localización de robot y otros problemas paralelos de estimación de marcas independientes, la complejidad computacional se reduce a lineal en el número de marcas de referencia y poses n , pudiendo lograr una complejidad logarítmica al usar una estructura de datos adecuada. Sin embargo, también depende linealmente del número de partículas k utilizadas para el filtro, lo que produce un tiempo de ejecución de $\mathcal{O}(k \log n)$.

Casi simultáneamente Thrun et. al [44, 45] también proponen utilizar el filtro de información extendida (EIF), análogo al EKF pero utilizando una parametrización gaussiana en forma de información [46]. La matriz de información (de Fisher), re-

lacionada por su inversa a la matriz de covarianza tradicional, es bastante rala en comparación a esta última. La desactivación (una heurística de marginalización de la medición propuesta en este documento) cuidadosa de marcas de referencia distantes explota aún más esta característica mediante la imposición de un límite constante en el número de características activas consideradas por las actualizaciones de medición. Por lo tanto, en cada actualización solo se necesita actualizar un número constante de entradas relacionadas en la matriz de información para cada variable.

Motivado por los resultados recientes en fotogrametría [47], y el hecho de que el filtrado fue demostrado inconsistente cuando es aplicado al problema inherentemente no lineal de SLAM [48], Dellaert y Kaess [16] están entre los primeros en sugerir una solución de mínimos cuadrados para el problema completo de SLAM mediante el uso del algoritmo de *Bundle Adjustment* (BA) para optimizar la trayectoria del robot y las marcas de referencia del entorno simultáneamente. Esto es análogo a la aplicación iterativa del paso de actualización de EKF varias veces, aunque se vuelven a calcular los puntos de linealización hasta lograr la convergencia. Aunque se aplica una heurística de dispersión para mejorar el rendimiento, el problema completo es considerado para la optimización, lo que conlleva un aumento potencialmente ilimitado en la complejidad. Sin embargo, la optimización de todo el problema, también denominado *Bundle Adjustment global* (GBA), sigue siendo el estándar contra el que se comparan todas las heurísticas derivadas para evaluar su desempeño.

Construyendo sobre la idea de *Bundle Adjustment global*, Mouragnon et al. [49] introducen la noción de *Bundle Adjustment local* (LBA) en un esfuerzo por lograr un rendimiento en tiempo real para la optimización del mapa. En ella solo un número constante de los fotogramas clave más recientes son considerados en la optimización, junto con todas las marcas de referencia proyectadas en las imágenes correspondientes. Un conjunto constante de otros fotogramas clave adyacentes se mantiene fijo dentro del optimizador para mantener restricciones fuertes sobre las marcas de referencia optimizadas. La incertidumbre de la posición actual se estima utilizando la inversa de la matriz hessiana del problema. Esto representa aproximadamente la covarianza de la distribución condicional sobre los parámetros excluidos, pero se vuelve inconsistente en términos globales, lo que hace que la información sea inútil para tareas como la asociación o planificación activa de datos.

Esto es señalado por Eudes y Lhuillier [50], quienes sostienen que el LBA original no produce un estimador global consistente, ya que las covarianzas de los parámetros de omisión se ignoran y solo se considera el ruido de medición local en cada paso. Se propone un Estimador de Probabilidad Máxima (MLE) en tiempo real para las poses optimizadas, aunque bajo el fuerte supuesto de independencia con respecto a las características observadas. Esto se logra estimando las derivadas de la función de minimización para calcular aproximaciones de primer orden de las incertidumbres de la pose. El estimador para la covarianza propaga efectivamente el ruido acumulado presente en las matrices de covarianza calculadas recursivamente de los fotogramas clave fijos.

Konolige [51] parece ser, a conocimiento del autor, el primero en explicar la reducción de la complejidad en SLAM en términos de marginalización probabilística. Al tratar los elementos del problema (poses, marcas de referencia y mediciones) como variables aleatorias condicionalmente independientes, la poda de variables puede modelarse como marginalización sobre la distribución conjunta del problema de Bundle Adjustment global. Esto produce una solución más consistente al problema de minimización en comparación con las heurísticas mencionadas anteriormente, que son equivalentes al condicionamiento de variables. La marginalización se realiza a nivel de la matriz de información utilizando el complemento de Schur para lograr una reducción de variables. Además, se muestra que las soluciones más populares de SLAM a esa fecha como EKF/EIF [44], CPE [52, 53] y FastSLAM [42] se pueden pensar como casos particulares de marginalización sobre la distribución conjunta original. Sin embargo, también se observa que estas operaciones inducen dependencias probabilísticas entre variables condicionalmente independientes. Esto conlleva una densificación del grafo de restricciones y, por extensión, también del problema de optimización. En resumen, el número de variables en el problema se reduce, pero el número de restricciones crece, aunque la mayoría de los métodos se basan en heurísticas de ralización para mitigar este problema.

Usando la misma técnica de marginalización, Sibley et al. [54, 55, 56] proponen un filtro basado en una ventana deslizante de tamaño fijo en el que las poses más antiguas que quedan fuera de la ventana se marginalizan de una manera probabilísticamente consistente, garantizando así la misma solución que el Bundle Adjustment global (exceptuando errores de linealización), aunque en tiempo constante. En cada paso se actualiza un sub-bloque (de tamaño fijo) de la matriz hessiana en cada paso que modela las restricciones inducidas por las poses eliminadas sobre los datos optimizados. En el contexto de aproximación y aterrizaje planetarios, donde no son posibles los ciclos en la trayectoria, éste enfoque es ideal.

Eade [57] también describe la eliminación de nodos y la poda de aristas para un grafo de SLAM como una forma de reducir la complejidad del problema original. Como en esta presentación del problema todas las aristas entre nodos representan esencialmente transformaciones inciertas en el espacio euclidiano, los nuevos valores para las aristas resultantes de la marginalización se calculan al componer y promediar las transformaciones de manera intuitiva. El algoritmo de marginalización se establece al nivel del grafo de restricciones, introduciendo y actualizando las aristas entre los nodos, que representan los parámetros del problema, en lugar de manipular directamente la matriz de información del problema. Además, se propone una heurística de esparsificación que mantiene constante el grado de cada nodo descartando las mediciones con los residuos más bajos mientras no comprometan la conectividad del grafo. Sin embargo, al seleccionar las mediciones descartadas de esta manera, se introduce un sesgo en el sistema. Además, solo se presenta una solución de marginalización para las restricciones de rango completo entre las poses, dejando de lado la marginalización de las restricciones del grafo que representan factores genéricos y de rango potencialmente incompleto (por ejemplo, mediciones de marcas de referencia

en una imagen).

Carlone et. al [58] presentan una novedosa heurística para aproximar la verdadera matriz de información marginal de una ventana mediante la introducción arbitraria de una serie de factores inteligentes entre las variables restantes. Estos factores tienen una estructura arbitrariamente fija de información previa, y sus valores se optimizan de manera que aproxime lo mejor posible a la verdadera matriz de información. No obstante, la matriz de información original aún debe calcularse para calcular los residuos de la función de optimización.

Recientemente, Keivan y Sibley [59] abandonan la estrategia de marginalización, ya que densifica la matriz, arrastra errores de linealización y requiere una re-marginalización global muy costosa durante los cierres de ciclos. Con el fin de condicionar las partes inactivas del mapa, se propone una estrategia para seleccionar incrementalmente los fotogramas clave que se optimizarán hasta que los residuos observados puedan explicarse por las covarianzas locales computadas. Sin embargo, no se mantiene una covarianza coherente a nivel global, y aunque el tamaño de la ventana no crece más allá de un tamaño razonable en los experimentos presentados, no se puede dar ninguna garantía al respecto en el caso general.

Kopitkov e Indelman [60] por otro lado, mientras siguen apostando a la marginalización de los parámetros dejados de lado con todos los problemas mencionados anteriormente, proponen una novedosa estrategia de selección de ventanas que maximiza la ganancia de información. Aunque costoso de computar en la práctica, la complejidad se puede lograr en tiempo constante.

3 Preliminares

3.1. Convenciones de notación

- Variables en negrita y minúscula (\mathbf{x}) denotan vectores.
- Variables en negrita y mayúscula (\mathbf{X}) denotan matrices.
- Variables sin negrita y en minúscula denotan escalares (x).
- Variables sin negrita y en mayúscula denotan conjuntos (X).
- Un superíndice por delante indica el marco de referencia de un objeto: ${}^W \mathbf{x}$.
- Una matriz de transformación se notará apropiadamente como ${}^B_A \mathbf{T}$ y transforma vectores del marco de referencia A al marco de referencia B : ${}^B \mathbf{x} = ({}^B_A \mathbf{T}) {}^A \mathbf{x}$.
- $p(\mathbf{x})$ denota la función de densidad de probabilidad (PDF) de una variable aleatoria (v.a.) \mathbf{x} .
- $p(\mathbf{x}|\mathbf{y})$ denota la PDF condicional de una v.a. \mathbf{x} dada otra v.a. \mathbf{y} .
- $x \sim \mathcal{N}(\mu, \sigma^2)$ denota una v.a. x que sigue una distribución normal (gaussiana) con media μ y desvío estándar σ .
- Análogamente $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denota una v.a. normal multivariada con media $\boldsymbol{\mu} \in \mathbb{R}^n$ y matriz de varianza-covarianza $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$, donde $\boldsymbol{\Sigma}$ siempre es semidefinida positiva.
- Dadas dos muestras aleatorias $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ de la misma distribución con matriz de varianza-covarianza $\mathbf{C} \in \mathbb{R}^{n \times n}$, $\|\mathbf{a} - \mathbf{b}\|_{\mathbf{C}}$ denota la distancia de Mahalanobis entre \mathbf{a} y \mathbf{b} , que se define como $\sqrt{(\mathbf{a} - \mathbf{b})^\top \mathbf{C}^{-1} (\mathbf{a} - \mathbf{b})}$.

3.2. Distribución normal multivariada

Muchos de los modelos probabilísticos descritos en el documento actual generalmente se basan en el supuesto de que las variables siguen una distribución normal multivariada. Además, la partición de densidades multivariadas grandes será útil para concentrarse de manera más eficiente en subconjuntos de variables particularmente interesantes. La forma más común de parametrizar una distribución normal

es a través de la forma del *momentos*

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \triangleq \frac{1}{(2\pi)^{n/2} \sqrt{\det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right), \quad (3.1)$$

donde \mathbf{x} denota una distribución normal multivariada de media $\boldsymbol{\mu} \in \mathbb{R}^n$ y matriz de varianza-covarianza $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$. Una parametrización alternativa está dada por la forma *natural* o de *información*

$$\mathcal{N}^{-1}(\mathbf{x}; \boldsymbol{\nu}, \boldsymbol{\Lambda}) \triangleq \frac{\exp\left(-\frac{1}{2}\boldsymbol{\nu}^\top \boldsymbol{\Lambda}^{-1}\boldsymbol{\nu}\right)}{(2\pi)^{n/2} \sqrt{\det \boldsymbol{\Lambda}^{-1}}} \exp\left(-\frac{1}{2}\mathbf{x}^\top \boldsymbol{\Lambda} \mathbf{x} + \mathbf{x}^\top \boldsymbol{\nu}\right), \quad (3.2)$$

donde $\boldsymbol{\nu} \in \mathbb{R}^n$ es llamado vector de información y $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$ la matriz de información (de Fisher).

Por manipulación algebraica de ambas distribuciones, es fácil notar que

$$\boldsymbol{\nu} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \quad (3.3)$$

y

$$\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}. \quad (3.4)$$

Ambas $\boldsymbol{\Lambda}$ y $\boldsymbol{\Sigma}$ son matrices simétricas semi-definidas positivas.

3.2.1. Normal multivariada particionada

Dada una variable aleatoria \mathbf{x} que sigue una distribución normal multivariada $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, puede ser particionada sin pérdida de generalidad como $\mathbf{x} = \begin{bmatrix} \mathbf{x}_a & \mathbf{x}_b \end{bmatrix}$ con

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ab}^T & \boldsymbol{\Sigma}_{bb} \end{bmatrix} \quad (3.5)$$

o equivalentemente

$$\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_a \\ \boldsymbol{\nu}_b \end{bmatrix} \quad \boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ab}^T & \boldsymbol{\Lambda}_{bb} \end{bmatrix}. \quad (3.6)$$

Es fácil derivar la siguiente relación

$$\boldsymbol{\Lambda}_{aa} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ab}^T, \quad (3.7)$$

e donde se deducen los siguientes teoremas:

Teorema 3.1 (Marginalización). La distribución marginal de \mathbf{x}_a está dada por

$$\mathbf{x}_a \sim \mathcal{N}(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa}) \quad (3.8)$$

o alternativamente

$$\mathbf{x}_a \sim \mathcal{N}^{-1}\left(\boldsymbol{\nu}_a - \boldsymbol{\Lambda}_{ab}\boldsymbol{\Lambda}_{bb}^{-1}\boldsymbol{\nu}_b, \boldsymbol{\Lambda}_{aa} - \boldsymbol{\Lambda}_{ab}\boldsymbol{\Lambda}_{bb}^{-1}\boldsymbol{\Lambda}_{ab}^\top\right), \quad (3.9)$$

donde $\boldsymbol{\Lambda}_{aa} - \boldsymbol{\Lambda}_{ab}\boldsymbol{\Lambda}_{bb}^{-1}\boldsymbol{\Lambda}_{ab}^\top$, también notado $\boldsymbol{\Lambda}/\boldsymbol{\Lambda}_{bb}$, se denomina el *complemento de Schur* de $\boldsymbol{\Lambda}_{bb}$ en $\boldsymbol{\Lambda}$.

Teorema 3.2 (Condicionamiento). La distribución condicional de \mathbf{x}_a dado un valor fijo \mathbf{x}_b^* para \mathbf{x}_b está dada por

$$(\mathbf{x}_a|\mathbf{x}_b = \mathbf{x}_b^*) \sim \mathcal{N}\left(\boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b^* - \boldsymbol{\mu}_b), \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ab}^\top\right) \quad (3.10)$$

o alternativamente

$$(\mathbf{x}_a|\mathbf{x}_b = \mathbf{x}_b^*) \sim \mathcal{N}^{-1}(\mathbf{v}_a - \boldsymbol{\Lambda}_{ab}\mathbf{v}_b, \boldsymbol{\Lambda}_{aa}). \quad (3.11)$$

3.2.2. Log-verosimilitud

A veces resulta útil operar con expresiones más simples en lugar de funciones explícitas de densidad normal, que aún conservan algunas propiedades requeridas. Uno de estos casos se encuentra al realizar una optimización sobre los valores medios de la distribución. Considere una función de densidad normal para una variable aleatoria $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$p(\mathbf{x}) = \eta \exp\left(-\frac{1}{2}\|\mathbf{x} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}}^2\right) \quad (3.12)$$

donde η es una constante de normalización que no depende de $\boldsymbol{\mu}$. La *log-verosimilitud* de \mathbf{x} se define como

$$\ln(p(\mathbf{x})) = \ln(\eta) - \frac{1}{2}\|\mathbf{x} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}}^2. \quad (3.13)$$

Dado que el término $\ln(\eta)$ no depende del valor de la media $\boldsymbol{\mu}$, puede ser ignorado sin consecuencias sobre la optimización, y la expresión resultante

$$\frac{1}{2}\|\mathbf{x} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}}^2 \quad (3.14)$$

se vuelve una función lineal sobre el parámetro $\boldsymbol{\mu}$.

3.2.3. Intervalos de confianza

El modelado de intervalos de confianza para distribuciones normales multivariadas será útil por dos razones: por un lado, se pueden usar en algoritmos de asociación de datos para verificar si algunos datos dados son bien descritos por algún modelo probabilístico hipotético, y por el otro se pueden usar para proporcionar representaciones visuales intuitivas de las distribuciones estimadas para algunas de las variables modeladas en el problema.

Dada una variable multivariada \mathbf{x} que sigue una distribución normal $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ con $\boldsymbol{\mu} \in \mathbb{R}^k$ y $\boldsymbol{\Sigma} \in \mathbb{R}^{k \times k}$. La expresión

$$\|\mathbf{x} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}^{-1}}^2 = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \quad (3.15)$$

sigue una distribución chi-cuadrado con k grados de libertad [61], denotada χ_k^2 .

Esto permite [62] definir un umbral $\chi_{k,\alpha}^2$ para un intervalo de confianza arbitrario tomando la inversa de la distribución acumulada de χ_k^2 hasta el nivel de significancia α ($0 \leq \alpha < 1$) deseado, a veces llamado *compuerta de validación*. Una muestra $\check{\mathbf{x}}$ de \mathbf{x} es marcada como un valor atípico si

$$\chi_{k,\alpha}^2 < \|\check{\mathbf{x}} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}}^2. \quad (3.16)$$

La probabilidad de que esto sea un falso negativo es de $1 - \alpha$. Los valores correspondientes para $\chi_{k,\alpha}^2$ para valores habituales se pueden encontrar en libros de texto de estadística. Valores habituales para α suelen ser 0,95 o 0,99.

Además, la ecuación general para puntos \mathbf{a} dentro de un elipsoide centrado alrededor de un punto \mathbf{v} es

$$(\mathbf{a} - \mathbf{v})^T \mathbf{A} (\mathbf{a} - \mathbf{v}) \leq s \quad (3.17)$$

siendo \mathbf{A} una matriz positiva-definida y s un parámetro de escala para el volumen. Por analogía con 3.16, queda claro que los intervalos de confianza para distribuciones normales multivariadas de tamaño k pueden representarse con elipsoides en el espacio k -dimensional. Por lo tanto, los intervalos de confianza para una distribución normal multivariada se pueden visualizar como un elipsoide parametrizado de la siguiente manera

$$(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq \chi_{k,\alpha}^2 \quad (3.18)$$

donde cualquier punto dentro de dicho elipsoide cae dentro del intervalo de confianza elegido para la distribución.

3.3. Propagación de la incertidumbre

Las muestras de variables que modelan mediciones físicas del mundo real están destinadas a sufrir cierto grado de incertidumbre debido a limitaciones en los sensores. Suponiendo que la variable sigue una cierta distribución estadística, estos errores se pueden modelar y cuantificar en términos del desvío estándar. Además, si se aplicaran funciones sobre estas variables aleatorias, puede resultar interesante propagar esta incertidumbre al resultado a través de las mismas.

Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $f = \mathbf{A}\mathbf{x} + \mathbf{b}$ una función lineal de una variable aleatoria multivariada \mathbf{x} con matriz de varianza-covarianza $\Sigma_{\mathbf{x}} \in \mathbb{R}^{n \times n}$. Luego la matriz de varianza-covarianza Σ_f está dada por

$$\Sigma_f = \mathbf{A}\Sigma_{\mathbf{x}}\mathbf{A}^T. \quad (3.19)$$

Si f fuese una función no-lineal de \mathbf{x} , hay pocos casos en los que es posible derivar una fórmula exacta para la propagación de incertidumbre. Por lo tanto es común emplear una aproximación por serie de Taylor de primer orden

$$f \approx f_0 + \mathbf{J}\mathbf{x}, \quad (3.20)$$

donde \mathbf{J} es la derivada de primer orden de f con respecto a \mathbf{x} , usualmente referida como la matriz *Jacobiana* de f . Como esta ecuación es lineal y f_0 constante, por lo cual no contribuye al error, la propagación puede ser aproximada por

$$\Sigma_f \approx \mathbf{J}\Sigma_{\mathbf{x}}\mathbf{J}^T. \quad (3.21)$$

Es importante tener en cuenta que la propagación a través de funciones no lineales de esta manera produce estimadores sesgados para la covarianza. La extensión de este sesgo depende del grado de no linealidad de la función.

3.4. Transformaciones inciertas

Muchas de las operaciones involucradas en el contexto de SLAM continuamente hacen uso de transformaciones de coordenadas y vectores en el espacio euclidiano 3D de un marco de referencia a otro. Al considerar un enfoque probabilístico, la mayoría de estos vectores y marcos de referencia modelan poses inciertas de robots, marcas de referencia del ambiente o sensores, por lo que las transformaciones involucradas conllevan un cierto grado de incertidumbre. Esta incertidumbre se puede propagar a los objetos transformados, proporcionando información útil que puede ser explotada por otros algoritmos espaciales (por ejemplo, filtros de fusión, asociación activa, marginalización basada en información, detección y corrección de ciclos, etc.). Se presenta a continuación una parametrización de dichos objetos y el álgebra correspondiente que permitirá un diseño simple y altamente cohesivo.

3.4.1. Parametrización de transformaciones rígidas

El conjunto de transformaciones rígidas que pueden ser realizadas en un espacio euclídeo n -dimensional forman un grupo algebraico denominado el *Grupo Euclídeo Especial* ($SE(n)$). El número de grados de libertad de los objetos pertenecientes al grupo es $n(n+1)/2$, donde n pueden ser atribuidos al subgrupo de *simetría traslacional*, notado $T(n)$, y los restantes $n(n-1)/2$ al subgrupo de *simetría rotacional*, representado por el grupo *especial ortogonal* $SO(3)$. Para $SE(3)$, el conjunto de transformaciones rígidas en el espacio tridimensional, esto equivale a 6 grados de libertad: 3 para las traslaciones y 3 para las rotaciones. Ambos $SO(3)$ y $SE(3)$ también son conocidos por formar *grupos de Lie* compactos de dimensiones 3 y 6 respectivamente.

Cada grupo de Lie tiene asociada un álgebra de Lie, que se define como el espacio de vector tangente alrededor del elemento de identidad del grupo. La importancia de un álgebra de Lie reside en que proporciona un espacio óptimo para representar los elementos del grupo, ya que el número de dimensiones del espacio vectorial coincide con el número de grados de libertad de los elementos del grupo. Además, se puede definir una relación que asigne elementos de un espacio a su representación análoga en el otro. Esto permite que los cálculos fáciles y lineales en el espacio del grupo de Lie se transformen a su representación óptima en el espacio tangente. Además, la matriz *adjunta* transforma linealmente los elementos en el espacio del álgebra de Lie de un marco de referencia a otro, lo que resulta útil cuando se opera con transformaciones inciertas. Las álgebras de Lie para los grupos $SO(3)$ y $SE(3)$ serán llamados $so(3)$ y $se(3)$ respectivamente.

El grupo $SE(3)$ puede ser representado por transformaciones lineales sobre vectores homogéneos en 4D

$$\begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \in SE(3) \quad (3.22)$$

donde $R \in SO(3)$ y $t \in \mathbb{R}^3$. Por otro lado, el álgebra de Lie $se(3)$ puede ser parametrizada como

$$\boldsymbol{\mu} = \begin{pmatrix} \mathbf{u} & \boldsymbol{\omega} \end{pmatrix}^T \in se(3) \quad (3.23)$$

donde $u \in$ y $\omega \in se(3)$.

El *mapa exponencial* $\exp : se(3) \rightarrow SE(3)$ es la operación que convierte a una matriz antisimétrica en el espacio de álgebra de Lie a su matriz de rotación correspondiente en el espacio de grupos de Lie. Análogamente, el *mapa logarítmico* $\ln : SE(3) \rightarrow se(3)$ representa a la operación inversa. La matriz adjunta Adj_C de $C \in SE(3)$ satisface

$$C \cdot \exp(\boldsymbol{\mu}) = \exp(Adj_C \cdot \boldsymbol{\mu}) \cdot C \quad (3.24)$$

3.4.2. Incertidumbre en el espacio euclídeo tridimensional

Las incertidumbres para coordenadas, vectores y transformaciones se modelarán siguiendo una distribución normal en todos los casos. Para coordenadas y vectores, la incertidumbre asociada se representa mediante una matriz de varianza-covarianza en el espacio $\mathbb{R}^{3 \times 3}$. Para las transformaciones, las matrices análogas se representarán en se (3), produciendo una matriz de varianza-covarianza en $\mathbb{R}^{6 \times 6}$. Además, la operación de transformación sobre grupos de Lie debe extenderse para agregar correctamente las incertidumbres involucradas en el proceso.

3.4.2.1. Transformando un vector

Dado un punto incierto en el espacio euclidiano tridimensional con distribución $\mathcal{N}({}^A\mathbf{X}, {}^A\Sigma_{\mathbf{X}})$ y una transformación incierta en el mismo espacio con distribución $\mathcal{N}({}^B\mathbf{T}, {}^B\Sigma)$. El punto puede ser transformado al marco de referencia B con distribución resultante $\mathcal{N}({}^B\mathbf{X}, {}^B\Sigma_{\mathbf{X}})$ donde

$${}^B\mathbf{x} = {}^B\mathbf{R} \cdot {}^A\mathbf{x} + {}^B\mathbf{t} \quad (3.25)$$

$${}^B\Sigma_{\mathbf{X}} = ({}^B\mathbf{X}^\odot) \cdot {}^B\Sigma \cdot ({}^B\mathbf{X}^\odot)^T + {}^B\mathbf{R} \cdot {}^A\Sigma_{\mathbf{X}} \cdot {}^B\mathbf{R}^T \quad (3.26)$$

donde ${}^B\mathbf{T} = \begin{bmatrix} {}^B\mathbf{R} & {}^B\mathbf{t} \end{bmatrix}$ y el operador \odot está definido como en [63]

$$\mathbf{x}^\odot = \begin{bmatrix} \mathbf{I}_3 & -\mathbf{x}^\wedge \end{bmatrix} \quad (3.27)$$

donde \mathbf{x}^\wedge se define como la matriz simétrica oblicua que representa al producto cruzado

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}. \quad (3.28)$$

3.4.2.2. Concatenando transformaciones

Dadas dos transformaciones inciertas con distribuciones $\mathcal{N}({}^B\mathbf{T}, {}^B\Sigma)$ y $\mathcal{N}({}^C\mathbf{T}, {}^C\Sigma)$, la composición $\mathcal{N}({}^C\mathbf{T}, {}^C\Sigma)$ puede computarse como

$${}^C\mathbf{T} = {}^C\mathbf{T} \cdot {}^B\mathbf{T} \quad (3.29)$$

$${}^C\Sigma = {}^B\Sigma + \text{Adj}({}^B\mathbf{T}) \cdot {}^C\Sigma \cdot \text{Adj}({}^B\mathbf{T})^T \quad (3.30)$$

3.4.2.3. Transformación inversa

Dada una transformación incierta con distribución $\mathcal{N}\left(\begin{smallmatrix} B \\ A \end{smallmatrix} \mathbf{T}, \begin{smallmatrix} B \\ A \end{smallmatrix} \Sigma\right)$, su transformación inversa está dada por

$$\begin{smallmatrix} A \\ B \end{smallmatrix} \mathbf{T} = \left[R^T - R^T t \right] \quad (3.31)$$

$$\begin{smallmatrix} A \\ B \end{smallmatrix} \Sigma = \text{Adj} \left(\begin{smallmatrix} A \\ B \end{smallmatrix} \mathbf{T} \right) \begin{smallmatrix} B \\ A \end{smallmatrix} \Sigma \text{Adj} \left(\begin{smallmatrix} A \\ B \end{smallmatrix} \mathbf{T} \right)^T \quad (3.32)$$

Para mayor referencia sobre el tema, el lector es referido a [64] y [63]. En la práctica, los elementos y las operaciones sobre transformaciones rígidas inciertas en este trabajo se implementan extendiendo la biblioteca [65] para operar con relaciones inciertas bajo la asunción de un modelo de incertidumbre gaussiano.

3.5. Métricas de error en SLAM

Ha sido el consenso durante algún tiempo [66, 67] que juzgar el desempeño de los algoritmos SLAM basándose únicamente en las mediciones de error de pose absolutas tiene varios inconvenientes. La localización en SLAM se realiza en relación a un mapa computado incrementalmente, y diferentes mapas pueden producir diferentes geometrías o sesgos, incluso cuando se ejecuta el mismo algoritmo SLAM sobre diferentes conjuntos de datos. Dependiendo de la aplicación, estas diferencias pueden no ser relevantes siempre que las estimaciones sean coherentes a nivel local, lo que puede ser cierto a pesar de un error de pose absoluto elevado. Para hacer frente a este problema, generalmente se emplea otra métrica [66, 67] específicamente diseñada para evaluar el rendimiento de los sistemas SLAM.

La métrica de la raíz del error medio cuadrado (RMSE) para una secuencia de poses estimadas $\mathbf{x}_1 \dots \mathbf{x}_m$ y correspondientes verdades de terreno $\mathbf{x}_1^* \dots \mathbf{x}_m^*$ se define como

$$\epsilon_{\text{abs}} = \sqrt{\frac{1}{m} \sum_{k=1}^m (\mathbf{x}_k \ominus \mathbf{x}_k^*)^2} \quad (3.33)$$

donde \ominus es la inversa del operador de composición de transformaciones estándar \oplus [39]. Si hubiese información de incertidumbre disponible, también se la puede tener en cuenta [68] como

$$\epsilon_{\text{abs}} = \sqrt{\frac{1}{m} \sum_{k=1}^m (\mathbf{x}_k \ominus \mathbf{x}_k^*)^T \Sigma_k^{-1} (\mathbf{x}_k \ominus \mathbf{x}_k^*)} \quad (3.34)$$

donde Σ_k representa la matriz de covarianza marginal para la pose \mathbf{x}_k . Es importante tener en cuenta que han de estar disponibles poses de referencia sincronizadas

temporalmente con las estimaciones de SLAM. Este no suele ser el caso, ya que los sistemas de medición de verdades de terreno generalmente dependen de equipos externos que capturan estimaciones de pose en momentos diferentes y a una velocidad diferente que el algoritmo SLAM. Sin embargo, las verdades de terreno pueden interpolarse para que coincidan con las marcas de tiempo requeridas.

Como se mencionó anteriormente, se ha afirmado que estas métricas absolutas son subóptimas [66] para comparar los resultados de algoritmos SLAM. Los mismos autores proponen una nueva métrica que considera la energía de deformación que se necesita para transferir la estimación a la verdad de terreno. Sea $\boldsymbol{\delta}_{i,j} = \mathbf{x}_j \ominus \mathbf{x}_i$ la diferencia (o movimiento) entre dos poses de la misma secuencia, y $\boldsymbol{\delta}_{i,j}^* = \mathbf{x}_j^* \ominus \mathbf{x}_i^*$ la cantidad análoga para la verdad del terreno correspondiente. El error relativo entre los marcos i y j se puede definir como $\boldsymbol{\delta}_{i,j} \ominus \boldsymbol{\delta}_{i,j}^*$, y la métrica como

$$\epsilon_{\text{rel}} = \sqrt{\frac{1}{m} \sum_{i,j} \left(\text{trans} \left(\boldsymbol{\delta}_{i,j} \ominus \boldsymbol{\delta}_{i,j}^* \right) + \text{rot} \left(\boldsymbol{\delta}_{i,j} \ominus \boldsymbol{\delta}_{i,j}^* \right) \right)} \quad (3.35)$$

para un conjunto de poses apareadas $\{(\mathbf{x}_i, \mathbf{x}_j)\}$, donde

$$\text{trans} \left(\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \right) = \|\mathbf{t}\|_2 \quad (3.36)$$

es el largo de la distancia lineal recorrida

$$\text{rot} \left(\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \right) = \arccos \left(\frac{\text{tr}(\mathbf{R}) - 1}{2} \right) \quad (3.37)$$

es la cantidad de rotación angular. La métrica original se modificó tomando la raíz cuadrada, lo que ayuda a interpretar los resultados numéricos, ya que las unidades de medida son las mismas que para los datos. Estas ecuaciones dejan intencionalmente abierta la elección de qué desplazamientos relativos $\boldsymbol{\delta}_{i,j}$ se incluyen en la métrica, ya que, según lo discutido por los autores originales, diferentes opciones destacarán diferentes propiedades de los datos. Esta métrica, promediada en diferentes tamaños de distancia lineal para las poses apareadas \mathbf{x}_i y \mathbf{x}_j es la métrica principal utilizada para clasificar los algoritmos de SLAM en el popular *benchmark* de datos de KITTI. [67].

El trabajo actual toma prestada la idea de error relativo, aunque con algunas modificaciones. Por un lado, los errores de traslación y rotación se presentarán por separado, ya que son completamente diferentes en naturaleza y tienen diferentes fuentes de error. Por otro lado, aunque el valor de la raíz cuadrática media para ambas cantidades puede ser útil para clasificar los algoritmos entre sí, se fomentará una comparación más cualitativa de los datos al proporcionar gráficos estadísticos como diagramas de caja, que destacan más propiedades de los datos, p.e. errores mínimos y máximos. Para resaltar la consistencia local del algoritmo [66], se elegirán

pares de poses espacialmente cercanas, específicamente de fotogramas consecutivos, para calcular las diferencias. Los errores relativos de traslación y rotación de una trayectoria, por lo tanto, se calculan como

$$\epsilon^{\text{trans}} = \left\{ \text{trans} \left(\boldsymbol{\delta}_{k,k+1} \ominus \boldsymbol{\delta}_{k,k+1}^* \right) \right\} \quad (3.38)$$

y

$$\epsilon^{\text{rot}} = \left\{ \text{rot} \left(\boldsymbol{\delta}_{k,k+1} \ominus \boldsymbol{\delta}_{k,k+1}^* \right) \right\} \quad (3.39)$$

respectivamente.

4 Modelos de incertidumbre en VSLAM

no podemos esperar recuperar el verdadero estado del mundo

— Dellaert y Kaess

Es un hecho ampliamente aceptado que tanto la detección como la interacción con el mundo real a través de medios físicos inevitablemente conlleva errores con diversos grados de incertidumbre. Estos errores pueden ser producto de varios factores. Generalmente los entornos son altamente impredecibles, ya que su dinámica completa no puede ser modelada completamente. Los sensores y actuadores, a su vez, están sujetos a limitaciones mecánicas y están contaminados por ruido, así como propensos a la degradación y el mal funcionamiento. Además, los datos se representan y manipulan en programas de software que utilizan representaciones discretas, aunque las fuentes de información podrían ser continuas.

En un mundo donde a los robots se les asignan tareas cada vez más críticas, es conveniente, si no necesario, estimar la incertidumbre con la mayor precisión posible para poder emitir respuestas lo suficientemente robustas y tomar decisiones que no comprometan la seguridad o los recursos en juego. Además, la incertidumbre también puede usarse como una herramienta para optimizar los recursos disponibles, como es el caso del uso de la *asociación activa* para reducir el área en la que se puede encontrar una característica local de imagen correspondiente a una marca del ambiente, o incluso usarla como una métrica para realizar un SLAM activo que mantenga al robot lo mejor localizado posible durante la ejecución de una tarea.

Por lo tanto, es una propiedad deseable de cualquier sistema de SLAM proporcionar, además de la localización del robot y el mapa de su entorno, el grado de incertidumbre que se puede esperar de los datos proporcionados. El consenso general en la literatura contemporánea [1, 69, 70] es tratar el mundo físico en términos de modelos probabilísticos, es decir, cada propiedad física sobre el mundo real debe modelarse como una variable aleatoria que sigue una cierta distribución probabilística. En el contexto de SLAM basado en características, estas distribuciones de probabilidad corresponden principalmente a la pose del robot y de las marcas de referencia del ambiente, así como a las medidas tomadas por los sensores involucrados, p.e. cámaras, codificadores, IMU's. Por supuesto, esto implica manipular e interpretar los datos de una manera probabilísticamente consistente.

4.1. Planteamiento del problema

El problema de Visual SLAM considerado en el presente trabajo se compone de, aunque no se limita a, los siguientes objetos:

- Un conjunto de poses de *cámaras* $\{\mathbf{x}_0 \dots \mathbf{x}_m\}$, donde $\mathbf{x}_j \in SE(3)$.
- Un conjunto de posiciones de *marcas de referencia del ambiente* $\{\mathbf{l}_0 \dots \mathbf{l}_n\}$, que se consideran estáticas, donde $\mathbf{l}_i \in \mathbb{R}^3$.
- Un conjunto de *mediciones* $\{\mathbf{z}_{i,j}\}$, donde $\mathbf{z}_{i,j}$ representa la posición en la imagen de la característica que representa a la marca \mathbf{l}_i al ser observada por la cámara con pose \mathbf{x}_j .
- Un conjunto de *vectores de control* $\{\mathbf{u}_j\}$ que impulsa el modelo de proceso entre estados, p.e. mediciones de velocidad.

Dado que las únicas cantidades observables del sistema $\{\mathbf{z}_{j,i}\}$ y $\{\mathbf{u}_j\}$ son intrínsecamente inexactas, principalmente debido a aproximaciones y dinámicas sin modelar, todos los elementos enumerados se tratan como variables aleatorias para tener en cuenta las incertidumbres asociadas. Además, se hará un supuesto fuerte (pero común) al afirmar que todos siguen una distribución gaussiana. Esto puede no ser muy preciso en el caso general [63] pero simplifica enormemente la fórmula cerrada de las densidades de probabilidad y las operaciones realizadas entre ellas.

Dichos objetos también están relacionados por las siguientes restricciones, que también suelen ser [39] modeladas convenientemente como procesos (potencialmente) no lineales corrompidos por ruido gaussiano:

- la función de transición o *modelo de proceso*

$$\mathbf{x}_j = f(\mathbf{x}_{j-1}, \mathbf{u}_j) + \mathbf{w}_j \quad (4.1)$$

- la función de observación

$$\mathbf{z}_{i,j} = h(\mathbf{l}_i, \mathbf{x}_j) + \mathbf{v}_{i,j} \quad (4.2)$$

donde $\mathbf{v}_{i,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{i,j})$ y $\mathbf{w}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_j)$ denotan ruido gaussiano de media cero para los procesos de observación y transición respectivamente. De estas ecuaciones se deducen los siguientes modelos para las densidades condicionales asociadas:

$$(\mathbf{x}_j | \mathbf{x}_{j-1}, \mathbf{u}_j) \sim \mathcal{N}(f(\mathbf{x}_{j-1}, \mathbf{u}_j), \mathbf{Q}_j) \quad (4.3)$$

$$(\mathbf{z}_{i,j} | \mathbf{l}_i, \mathbf{x}_j) \sim \mathcal{N}(h(\mathbf{l}_i, \mathbf{x}_j), \mathbf{R}_{i,j}). \quad (4.4)$$

La Figura 4.1 representa una instancia simple de un problema de SLAM de esta índole.

Estos modelos estocásticos se utilizarán para inferir el estado del robot y sus alrededores a partir de los datos recopilados por los sensores. En términos probabilísticos,

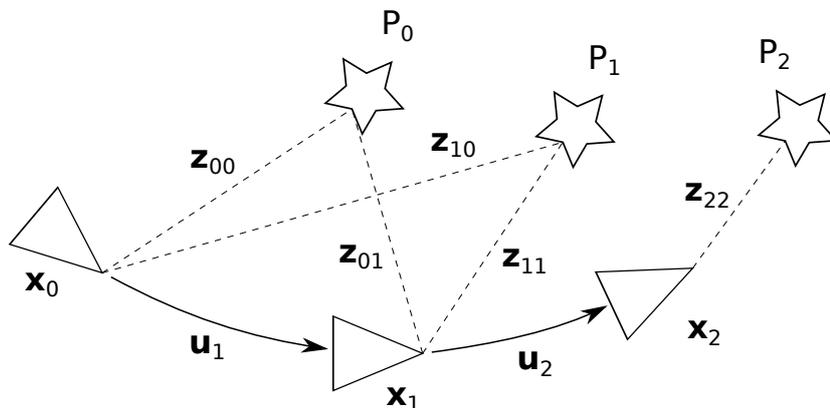


Figura 4.1: Representación espacial de un problema simple de SLAM. Los triángulos indican poses de robot que están relacionadas por restricciones de movimiento relativas. Las estrellas indican marcas de referencia del ambiente que están relacionados con poses través de mediciones de sensores, p.e. características de imagen o escaneos láser.

el problema de inferencia SLAM completo se puede definir como la estimación de la distribución de probabilidad de las variables desconocidas $X = \{\mathbf{x}_0 \dots \mathbf{x}_m, \mathbf{l}_0 \dots \mathbf{l}_n\}$ a partir de las variables observadas $Z = \{\mathbf{z}_{j,i}\} \cup \{\mathbf{u}_j\}$. El estimador más común de este tipo es el estimador máximo a posteriori (MAP)

$$\hat{X}^{MAP} = \arg \max_X p(X|Z). \quad (4.5)$$

llamado así porque maximiza la densidad posterior $p(X|Z)$ del estado X dadas las observaciones disponibles Z . Para calcular dicha optimización, se debe encontrar una fórmula cerrada para la función de costo dada $p(X|Z)$, lo cuál a primera vista no parece sencillo. Como se verá a continuación, el uso de *modelos gráficos probabilísticos* puede ayudar a definirla en términos de factores mas simples y de forma conocida.

4.2. Modelos gráficos

Los *modelos gráficos probabilísticos* [71] son una familia de representaciones gráficas diagramáticas de distribuciones de probabilidad. Esto se logra típicamente por medio de grafos, donde los nodos representan variables aleatorias y las aristas codifican relaciones probabilísticas entre ellas. Cuando se trata de distribuciones complica-

das de muchas dimensiones, estos diagramas proporcionan abstracciones poderosas que pueden resaltar la estructura subyacente y las propiedades condicionales de un modelo probabilístico. Además, muestran formas en las que la distribución de probabilidad conjunta de las variables puede ser factorizada en un producto de factores más pequeños de dimensionalidad reducida.

Existe una rica tradición de usar modelos gráficos para representar problemas de SLAM. A continuación se presenta un breve resumen de los más comunes. Para una discusión más amplia del tema, refiérase a [72].

4.2.1. Redes Bayesianas

Estos son probablemente los modelos gráficos más intuitivos típicamente asociados con el problema de SLAM. Una red bayesiana, también llamada red de creencias, es un grafo acíclico dirigido (DAG) donde los nodos representan variables aleatorias y las aristas representan relaciones causales directas entre ellas. La red es una representación gráfica de la densidad de probabilidad conjunta de todas las variables representadas.

Estas variables pueden dividirse además en *observables*, cuando es posible tomar muestras directas de la distribución modelada, o *latentes* (también llamadas ocultas), que modelan el proceso de generación subyacente de las primeras. En otras palabras, el modelo captura el proceso causal por el cual las variables latentes generan los datos observados.

Como la ausencia de aristas entre dos variables codifica una independencia condicional entre ellas, la distribución de probabilidad conjunta $p(\theta_1 \dots \theta_n)$ representada por una red bayesiana puede factorizarse como

$$p(\theta_1 \dots \theta_n) = \prod_{i=1}^n p(\theta_i | \text{padres de } \theta_i), \quad (4.6)$$

donde los padres de θ_i se definen como todos los nodos que tienen aristas directas en dirección a θ_i .

Un propósito común de estas redes es facilitar la formulación de la inferencia bayesiana, es decir, inferir las causas desconocidas de un modelo a partir de observar sus efectos. Por definición de probabilidad condicional, el problema de inferencia de SLAM presentado en la ecuación 4.5 se puede reformular como

$$\hat{X}^{MAP} = \arg \max_X p(X|Z) = \arg \max_X \frac{p(X, Z)}{p(Z)}. \quad (4.7)$$

Dado que el factor de normalización $p(Z)$ no depende de los parámetros X , se vuelve irrelevante para la optimización y se puede descartar de forma segura, dando como

resultado

$$\hat{X}^{MAP} = \arg \max_X p(X|Z) = \arg \max_X p(X, Z). \quad (4.8)$$

Esta formulación es mucho más conveniente que la declaración original, ya que se puede derivar una forma factorizada más simple para la probabilidad conjunta $p(X, Z)$ siguiendo la ecuación 4.6.

Tómese como ejemplo el pequeño problema de SLAM descrito en la Figura 4.1. La Figura 4.2 muestra la red bayesiana correspondiente. Esto permite que la densidad de probabilidad resultante del modelo pueda ser factorizada como

$$p(\{\mathbf{x}_j\} \cup \{\mathbf{l}_i\} \cup \{\mathbf{z}_{ij}\} \cup \{\mathbf{u}_j\}) = p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{u}_2) p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{u}_1) p(\mathbf{x}_0) \\ \times p(\mathbf{z}_{00}|\mathbf{x}_0, \mathbf{l}_0) p(\mathbf{z}_{01}|\mathbf{x}_1, \mathbf{l}_0) p(\mathbf{z}_{10}|\mathbf{x}_0, \mathbf{l}_1) p(\mathbf{z}_{11}|\mathbf{x}_1, \mathbf{l}_1) p(\mathbf{z}_{22}|\mathbf{x}_2, \mathbf{l}_2). \quad (4.9)$$

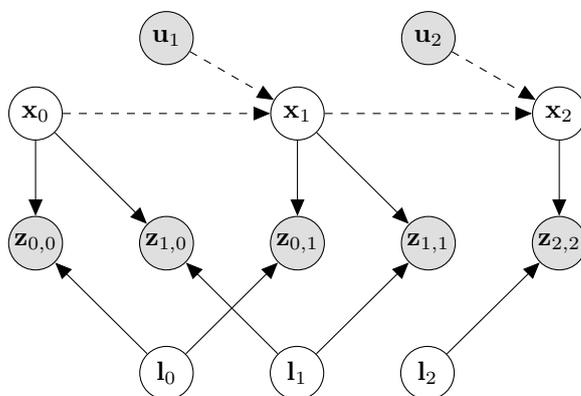


Figura 4.2: El problema de Visual SLAM en la Figura 4.1 representado como una Red Bayesiana. Los nodos blancos representan variables observables y los grises, latentes. Las medidas $\mathbf{z}_{i,j}$ indican una observación de la marca de referencia \mathbf{l}_i desde una pose \mathbf{x}_j . Las líneas de puntos indican las restricciones relacionadas con el modelo de proceso (ecuación 4.1) y las líneas aquellas relacionadas con la función de observación (ecuación 4.2).

Debido a la disposición típica de los elementos en el problema de SLAM, se pueden identificar fácilmente dos grupos distintos de factores:

- Una *Cadena de Markov* $p(\mathbf{x}_m|\mathbf{x}_{m-1}, \mathbf{u}_m) \dots p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{u}_1) p(\mathbf{x}_0)$ que sigue la trayectoria del robot.
- Un conjunto de densidades condicionales $p(\mathbf{z}_{ij}|\mathbf{x}_j, \mathbf{l}_i)$ resultado de las mediciones de marcas de referencia del ambiente realizadas por un sensor del robot en diferentes poses durante la trayectoria.

Todas las densidades que aparecen en la probabilidad conjunta factorizada para el problema SLAM son instancias de las densidades normales modeladas explícitamente en las ecuaciones 4.3 y 4.4. Por lo tanto, se puede dar una fórmula cerrada para la

función de costo $p(X, Z)$ que se puede usar para calcular una solución al problema de optimización requerido.

Sin embargo, se pueden aplicar manipulaciones adicionales a las funciones de densidad para simplificar el problema de optimización. Como consecuencia, la función de costo relacionada dejará de ser un producto de densidades probabilísticas. Esto, junto al hecho de que las redes bayesianas no suelen distinguir muy bien entre los parámetros del modelo y las observaciones dadas, motiva la introducción de un segundo lenguaje de modelado gráfico: el *grafo de factores*.

4.2.2. Grafos de factores

Los grafos de factores se pueden considerar como una generalización de las redes bayesianas y otros modelos gráficos probabilísticos, donde la función representada no es necesariamente una densidad probabilística: se pueden usar para especificar cualquier función factorizada sobre un conjunto de variables. Un grafo de factores se puede definir como un grafo bipartito con variables y factores. A su vez, un factor puede ser cualquier tipo de función definida entre las variables directamente adyacentes. Sea X el conjunto de variables en el grafo y $\{f_k\}$ el conjunto de factores. El grafo de factores describe la función

$$f(X) = \prod_k f_k(X_k), \quad (4.10)$$

donde X_k denota el conjunto de variables adyacentes al factor f_k .

4.2.2.1. Conversión a partir de redes bayesianas

Aunque las redes bayesianas son una gran herramienta para modelar el problema de SLAM y sus dependencias causales, los grafos de factores permiten representaciones más compactas y son más representativos del problema de inferencia asociado al grafo. Además, los algoritmos que realizan acciones interesantes como la inferencia y la marginalización de variables son más simples de definir sobre grafos de factores, a pesar de haber sido diseñados originalmente sobre redes bayesianas.

Cualquier red bayesiana puede traducirse fácilmente en un grafo de factores. Dada una red bayesiana BN y un grafo de factores vacío FG . Por cada nodo latente en BN se crea un nodo variable en FG . Por cada nodo con aristas entrantes en BN , se define un nodo factor con un valor correspondiente a la función de densidad de probabilidad condicional asociada y se conecta a las variables que tenían aristas directas hacia el nodo factorizado. Si el nodo factorizado era una variable latente, también se conecta a sí misma.

Tómese el ejemplo del problema de SLAM representado en la Figura 4.1 y la representación de red bayesiana correspondiente que se muestra en la Figura 4.2. Usando

el algoritmo mencionado anteriormente, es sencillo calcular el grafo de factores asociados que se muestra en la Figura 4.3.

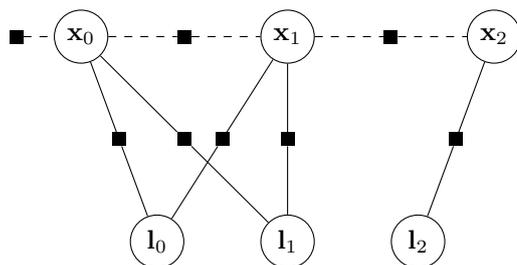


Figura 4.3: El problema de Visual SLAM de la Figura 4.1 representado como un grafo de factores. Para mayor claridad, las líneas de puntos indican las relaciones relacionadas con el modelo de proceso y las líneas completas las relacionadas con la función de observación.

Al condicionar los parámetros sobre el conjunto de mediciones observadas Z , siguiendo la ecuación 4.10, la factorización definida por el grafo de factores produce

$$p(X|Z) = f_1(\mathbf{x}_2, \mathbf{x}_1) f_2(\mathbf{x}_1, \mathbf{x}_0) f_3(\mathbf{x}_0) \\ \times f_4(\mathbf{x}_0, \mathbf{l}_0) f_5(\mathbf{x}_0, \mathbf{l}_1) f_6(\mathbf{x}_1, \mathbf{l}_0) f_7(\mathbf{x}_1, \mathbf{l}_1) f_8(\mathbf{x}_2, \mathbf{l}_2), \quad (4.11)$$

donde los factores f_k aún representan funciones de densidad de probabilidad. La correspondencia con los factores proporcionados por la factorización sobre una red bayesiana en la ecuación 4.9 debería ser fácil de notar.

4.3. Estimación de parámetros en SLAM

Al realizar la inferencia de mapa en problemas de SLAM para permitir el uso futuro del mismo, es conveniente mantener el mapa construido lo más consistente posible utilizando toda la información disponible, al mismo tiempo que se proporciona la localización en tiempo real. Las versiones contemporáneas de SLAM generalmente adoptan la arquitectura popularizada por PTAM [12], donde una tarea de *front-end* se dedica al seguimiento en tiempo real y una tarea de *back-end* se ejecuta en un hilo separado que optimiza porciones más grandes del mapa cuando se incorporan nuevas restricciones. Estas tareas serán referidas comúnmente como las operaciones de *seguimiento de pose* y *optimización de mapa* respectivamente. Normalmente, ambas tareas se logran mediante la optimización de la pose de la cámara y las posiciones de las marcas de referencia minimizando los errores cuadrados de medición. Este proceso es conocido como *Bundle Adjustment* [47].

Aunque la solución ideal para la optimización de mapa es optimizar las variables latentes usando toda la información disponible en cada paso, se hace imposible hacerlo en un tiempo de ejecución acotado si se espera que el mapa crezca de manera

ilimitada. Por lo tanto, las aproximaciones deben realizarse seleccionando partes del mapa de tamaño limitado para optimización y buenos momentos para realizarla. Sin embargo, la solución ideal, también conocida como la solución *batch*, se presenta a continuación como un punto de partida para inferir heurísticas más eficientes y también como un estándar para medir el rendimiento de dichos algoritmos.

4.3.1. Bundle Adjustment global

El problema *Bundle Adjustment* completo consiste en estimar el conjunto de poses del sensor y marcas de referencia $X = \{\mathbf{x}_j\} \cup \{\mathbf{l}_i\}$ dado el conjunto de las medidas observadas $Z = \{\mathbf{z}_{i,j}\} \cup \{\mathbf{u}_j\}$. Esencialmente pretende encontrar un buen estimador, según algún criterio de optimalidad, para el conjunto de parámetros latentes X relacionados entre sí a través de un conjunto de funciones de costo. Al considerar el enfoque bayesiano, uno de estos estimadores es el MAP presentado en Sección 4.2.1.

En este contexto, se puede derivar una factorización de la red bayesiana de dependencias causales asociadas al problema de Bundle Adjustment

$$\hat{X}^{MAP} = \arg \max_X \prod_{i=0}^n \prod_{j=0}^m p(\mathbf{z}_{i,j} | \mathbf{l}_i, \mathbf{x}_j) \prod_{j=1}^m p(\mathbf{x}_j | \mathbf{x}_{j-1}, \mathbf{u}_j) \prod_{i=0}^n p(\mathbf{l}_i) p(\mathbf{x}_0). \quad (4.12)$$

Además, generalmente se asume un prior uniforme sobre las marcas de referencia $\{\mathbf{l}_i\}$, dando como resultado

$$\hat{X}^{MAP} = \arg \max_X \prod_{i=0}^n \prod_{j=0}^m p(\mathbf{z}_{i,j} | \mathbf{l}_i, \mathbf{x}_j) \prod_{j=1}^m p(\mathbf{x}_j | \mathbf{x}_{j-1}, \mathbf{u}_j) p(\mathbf{x}_0). \quad (4.13)$$

Si se asumiera el mismo prior uniforme sobre la pose origen del grafo, la solución al problema de optimización no quedaría determinada de forma única: como todas las relaciones en el sistema son relativas, cualquier estimación práctica del sistema puede desplazarse en el espacio por la misma transformación de similitud sin afectar la función objetivo, generando el mismo grado de optimalidad y haciendo que el problema sea no observable. Este problema se conoce comúnmente como *gauge freedom* [47] y es resuelto considerando, sin pérdida de generalidad, la pose inicial \mathbf{x}_0 como condicionada a un valor arbitrario $\check{\mathbf{x}}_0$, o equivalentemente con un prior infinitamente preciso con media $\check{\mathbf{x}}_0$, también denominado origen del mapa. En la práctica $\check{\mathbf{x}}_0$ se toma generalmente como el origen del espacio euclidiano en el que se representan todos los objetos. A partir de aquí se omitirá la mención explícita de un prior sobre \mathbf{x}_0 para no entorpecer la legibilidad. La notación del parámetro X se relajará para abarcar el valor fijo \mathbf{x}_0 y las variables realmente optimizadas $X_{1:m}$

$$\hat{X}^{MAP} = \arg \max_X \prod_{i=0}^n \prod_{j=0}^m p(\mathbf{z}_{i,j} | \mathbf{l}_i, \mathbf{x}_j) \prod_{j=1}^m p(\mathbf{x}_j | \mathbf{x}_{j-1}, \mathbf{u}_j). \quad (4.14)$$

La forma explícita de estos factores se conoce a partir de las ecuaciones 4.3 y 4.4. Luego se puede calcular una solución para X utilizando un algoritmo de optimización numérica.

No es raro en la literatura de SLAM encontrar funciones objetivas similares para el estimador \hat{X} . En su planteo original [73] como un algoritmo de alineación de fotogrametría, normalmente se usa un estimador de máxima verosimilitud (ML) para encontrar el conjunto de parámetros \hat{X}^{ML} que maximiza la probabilidad de observar las mediciones Z

$$\hat{X}^{ML} = \arg \max_X p(Z|X). \quad (4.15)$$

Otros enfoques directamente instancian un posterior bayesiano

$$\hat{X} = \arg \max_X p(Z|X) p(X). \quad (4.16)$$

Al manipular la definición de la densidad de probabilidad conjunta, se llega fácilmente a las siguientes equivalencias

$$p(X, Z) = p(Z|X) p(X) = p(X|Z) p(Z), \quad (4.17)$$

que dado que $p(Z)$ no depende de X y que se supone un prior uniforme $p(X)$ sobre las variables latentes, implica que la maximización de cualquiera de las funciones de densidad mencionadas anteriormente sobre X produce el mismo estimador óptimo como el anteriormente derivado \hat{X}^{MAP} . Es decir, en el contexto de Bundle Adjustment, todos los estimadores presentados producen el mismo conjunto óptimo de parámetros \hat{X} .

4.3.1.1. Funciones de costo explícitas

Como se mencionó anteriormente, se deben proporcionar funciones de costo explícitas para el algoritmo de optimización de cuadrados mínimos no lineales para permitirle recalculer los errores de medición en cada paso.

Reemplazando las densidades gaussianas asumidas en las ecuaciones 4.3 y 4.4 en la ecuación 4.14 y tomando la probabilidad logarítmica negativa de los estimadores

$$\begin{aligned} \hat{X}^{MAP} = \arg \min_X & \frac{1}{2} \sum_{j=0}^m \sum_{i=0}^n \|z_{i,j} - h(\mathbf{l}_i, \mathbf{x}_j)\|_{\mathbf{R}_{i,j}}^2 \\ & + \frac{1}{2} \sum_{j=1}^m \|\mathbf{x}_j - f(\mathbf{x}_{j-1}, \mathbf{u}_j)\|_{\mathbf{Q}_j}^2, \end{aligned} \quad (4.18)$$

que expresa la función de costo para el problema de optimización en cuestión.

Los términos se pueden reorganizar en forma de matriz apilándolos en vectores columna, y sus pesos en matrices de bloques diagonales

$$\boldsymbol{\epsilon}_h = \begin{bmatrix} \mathbf{z}_{0,0} - h(\mathbf{l}_0, \mathbf{x}_0) \\ \vdots \\ \mathbf{z}_{n,m} - h(\mathbf{l}_n, \mathbf{x}_m) \end{bmatrix} \quad \mathbf{W}_h = \text{diag}(R_{0,0} \dots R_{n,m}) \quad (4.19)$$

$$\boldsymbol{\epsilon}_f = \begin{bmatrix} \mathbf{x}_1 - f(\mathbf{x}_0, \mathbf{u}_1) \\ \vdots \\ \mathbf{x}_m - f(\mathbf{x}_{m-1}, \mathbf{u}_m) \end{bmatrix} \quad \mathbf{W}_f = \text{diag}(\mathbf{Q}_1 \dots \mathbf{Q}_m). \quad (4.20)$$

Uniendo todo nuevamente como $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}_h^\top, \boldsymbol{\epsilon}_f^\top]^\top$ y $\mathbf{W} = \text{diag}(\mathbf{W}_h, \mathbf{W}_f)$ se obtiene la función de costo en forma matricial

$$\hat{X}^{MAP} = \arg \min_X \frac{1}{2} \boldsymbol{\epsilon}^T \mathbf{W}^{-1} \boldsymbol{\epsilon}. \quad (4.21)$$

Téngase en cuenta que no es necesario separar y agrupar los términos en la ecuación 4.18 para llegar a este resultado, sin embargo, presenta una noción intuitiva de la independencia de los factores que pertenecen a diferentes fuentes de información y cómo las familias de factores pueden ser fácilmente descartadas o incorporadas.

4.3.1.2. Resolviendo el problema canónico

Habiendo llegado a una representación factorizada y compacta para la función de costo, el siguiente paso es buscar un algoritmo para calcular realmente una solución. La mejor solución (en el sentido de cuadrados mínimos) para este conjunto de ecuaciones se puede encontrar usando las ecuaciones normalizadas para el algoritmo de Levenberg-Marquardt [74]

$$(\mathbf{J}^T \mathbf{W}^{-1} \mathbf{J}) \hat{\mathbf{x}} = \mathbf{J}^T \mathbf{W}^{-1} \boldsymbol{\epsilon}(\mathbf{x}), \quad (4.22)$$

donde \mathbf{J} se corresponde con el jacobiano de primer orden de la función optimizada

$$\mathbf{J} = \frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{x}}. \quad (4.23)$$

El término $\boldsymbol{\Lambda} = \mathbf{J}^T \mathbf{W}^{-1} \mathbf{J}$ también se llama matriz *hessiana* o matriz de información del sistema. Es sabido que la matriz de información $\boldsymbol{\Lambda}$ de un problema de SLAM es ampliamente esparsa, por lo cuál el cálculo del error de cuadrados mínimos se puede acelerar significativamente realizando multiplicaciones entre bloques no nulos. Sin embargo el problema de la estimación del estado completo se vuelve imposible de tratar en el largo plazo porque escala con una complejidad de $\mathcal{O}(m^3 + mn)$, donde m es el número de poses estimadas y n el número de marcas de referencia

observadas. Una forma de hacer frente a este problema es evitar la optimización de parámetros (poses de cámaras y posiciones de marcas de referencia) que casi no se verán afectados por los cambios ocurridos desde que se ejecutó la última optimización. Surgen dos problemas principales que deben resolverse: cómo seleccionar esos parámetros, que se tratará en Sección 5.1, y cómo ignorarlos sin perder demasiado de la información que llevan, que se aborda en Sección 5.2.

El uso de un enfoque probabilístico para el problema de estimación de estado agrega información valiosa, pero conlleva el costo de elegir los modelos de incertidumbre adecuados para las variables y procesos aleatorios involucrados. Como se introdujo en Sección 4.1, los elementos aleatorios son las pose de la cámara $\boldsymbol{\mu}_j \in \text{SE}(3)$ a lo largo del tiempo, las posiciones de las marcas de referencia $\mathbf{X}_i \in \mathbb{R}^3$ que conforman el mapa, las posiciones de características de imagen $\mathbf{z}_{ij} \in \mathbb{R}^2$ medidas en cada pose de la cámara y los vectores de control, que en este caso particular es solo la velocidad angular $\boldsymbol{\omega}_t \in \text{se}(3)$.

4.3.1.3. Distribución del posterior

Idealmente el propósito de realizar Bundle Adjustment es obtener un estimador óptimo para la distribución $p(X|Z)$. Como se indicó anteriormente, una opción óptima para el valor medio es el estimador a posteriori \hat{X}^{MAP} que se puede calcular en el contexto de SLAM (linealizado) realizando una optimización de cuadrados mínimos. Suponiendo un posterior con distribución normal, resta por definir un estimador para la covarianza o matriz de información de la misma.

Imitando el proceso utilizado para calcular la media óptima $X^* = \{\mathbf{l}_i^*, \mathbf{x}_j^*\}$, las funciones de costo se linealizan mediante una *expansión de Taylor* (de primer orden) alrededor de X^*

$$f(\mathbf{x}_j, \mathbf{u}_{j+1}) \approx f(\mathbf{x}_j^*, \mathbf{u}_{j+1}) + \left. \frac{\partial f}{\partial \{\mathbf{x}_j\}} \right|_{(\mathbf{x}_j - \mathbf{x}_j^*)} \quad (4.24)$$

$$h(\mathbf{l}_i, \mathbf{x}_j) \approx h(\mathbf{l}_i^*, \mathbf{x}_j^*) + \left. \frac{\partial h}{\partial \{\mathbf{l}_i, \mathbf{x}_j\}} \right|_{(\mathbf{l}_i - \mathbf{l}_i^*, \mathbf{x}_j - \mathbf{x}_j^*)} \quad (4.25)$$

Luego las densidades asociadas en las ecuaciones 4.3 y 4.4 se pueden aproximar como

$$(\mathbf{x}_{j+1} | \mathbf{x}_j, \mathbf{u}_{j+1}) \sim \mathcal{N} \left(f(\mathbf{x}_j^*, \mathbf{u}_{j+1}) + \left. \frac{\partial f}{\partial \{\mathbf{x}_j\}} \right|_{\mathbf{x}_j^*} (\mathbf{x}_j - \mathbf{x}_j^*), \mathbf{Q}_j \right) \quad (4.26)$$

$$(\mathbf{z}_{i,j} | \mathbf{l}_i, \mathbf{x}_j) \sim \mathcal{N} \left(h(\mathbf{l}_i^*, \mathbf{x}_j^*) + \left. \frac{\partial h}{\partial \{\mathbf{l}_i, \mathbf{x}_j\}} \right|_{\mathbf{l}_i^*, \mathbf{x}_j^*} (\mathbf{l}_i - \mathbf{l}_i^*, \mathbf{x}_j - \mathbf{x}_j^*), \mathbf{R}_{i,j} \right). \quad (4.27)$$

Apilando apropiadamente los términos se llega a

$$(Z|X) \sim \mathcal{N}(\mathbf{z}^* + \mathbf{J}(\mathbf{x} - \mathbf{x}^*), \boldsymbol{\Lambda}_Z^{-1}) \quad (4.28)$$

donde $\mathbf{z}^* = h(\mathbf{x}^*)$.

Por el teorema 3 de [46] la distribución de la probabilidad conjunta sobre todas las variables pasa a ser

$$(X, Z) \sim \mathcal{N} \left(\begin{bmatrix} x^* \\ z^* \end{bmatrix}, \begin{bmatrix} J^\top \mathbf{\Lambda}_Z J & -J^\top \mathbf{\Lambda}_Z \\ -\mathbf{\Lambda}_Z J & \mathbf{\Lambda}_Z \end{bmatrix} \right), \quad (4.29)$$

y aplicando el teorema 2 de [46] se llega a

$$(X|Z) \sim \mathcal{N} \left(\mathbf{x}^*, [J^\top \mathbf{\Lambda}_Z J]^{-1} \right). \quad (4.30)$$

Recordando la matriz hessiana $H = J^\top \mathbf{\Lambda}_Z J$ involucrada en el método de optimización de cuadrados mínimos (ecuación 4.21), se deduce que

$$(X|Z) \sim \mathcal{N} \left(\mathbf{x}^*, \mathbf{H}^{-1} \right). \quad (4.31)$$

4.3.2. Seguimiento de pose

El proceso de seguimiento se puede considerar como un caso especial del algoritmo de Bundle Adjustment presentado anteriormente. Por razones de eficiencia, solo será posible optimizar la pose actual \mathbf{x}_{n+1} con respecto al mapa existente $\{X; Z\}$. El mapa se aumentará momentáneamente con la nueva pose y medidas tomando $X_{m+1} = X \cup \{\mathbf{x}_{m+1}\}$ y $Z_{m+1} = Z \cup \{\mathbf{z}_{i,m+1}\}$. Por analogía con la sección anterior, un buen estimador es de nuevo el que maximiza la distribución de probabilidad resulta ser

$$\hat{\mathbf{x}}_{m+1}^{MAP} = \arg \max_{\mathbf{x}_{n+1}} p(X_{m+1}|Z_{m+1}), \quad (4.32)$$

que por manipulación análoga a la del capítulo anterior produce

$$\hat{\mathbf{x}}_{m+1}^{MAP} = \arg \max_{\mathbf{x}_{m+1}} p(Z_{m+1}|X_{m+1}) p(X_{m+1}) \quad (4.33)$$

y puede ser factorizada como

$$\hat{\mathbf{x}}_{m+1}^{MAP} = \arg \max_{\mathbf{x}_{m+1}} \prod_{i=0}^n \prod_{j=0}^{m+1} p(\mathbf{z}_{i,j} | \mathbf{l}_i, \mathbf{x}_j) \prod_{j=1}^{m+1} p(\mathbf{x}_j | \mathbf{x}_{j-1}, \mathbf{u}_j) \prod_{i=0}^n p(\mathbf{l}_i) p(\mathbf{x}_0). \quad (4.34)$$

Al eliminar todos los términos que no dependen del argumento \mathbf{x}_{m+1} , la función de costo para el proceso de seguimiento se convierte en

$$\hat{\mathbf{x}}_{m+1}^{MAP} = \arg \max_{\mathbf{x}_{m+1}} \prod_{i=0}^n p(\mathbf{z}_{i,m+1} | \mathbf{l}_i, \mathbf{x}_{m+1}) p(\mathbf{x}_{m+1} | \mathbf{x}_m, \mathbf{u}_{m+1}) \quad (4.35)$$

para cada l_i medido por \mathbf{x}_{m+1} , donde de nuevo se conoce la forma cerrada para todos los factores.

Particularmente en el caso del trabajo actual, el vector de control que impulsa el modelo de proceso se tomará como las mediciones de velocidad angular preintegradas detectadas por una unidad de medición inercial desde el momento del último fotograma clave \mathbf{x}_m . Se encontró que la velocidad lineal se comportaba de forma bastante errática, por lo tanto cualquier intento de estimarla mediante un modelo de movimiento suave (por ejemplo, una caída lineal) arrojó peores resultados que simplemente estimarla como cero.

Luego cada medición inercial se puede representar mediante una transformación incierta con desplazamiento lineal cero. La incertidumbre para cada medición se puede estimar buscando la relación de ruido del sensor en la hoja de datos proporcionada por el fabricante del sensor. Por lo general, este ruido se da como no correlacionado e idéntico por eje. Por lo tanto, para una velocidad angular detectada $\boldsymbol{\omega} \in \mathbb{R}^3$ con ruido de sensor η integrado en un paso de tiempo δt (la tasa de medición del sensor), el modelo de proceso para la pose estimada se puede expresar en términos de transformaciones euclidianas inciertas (Sección 3.4) como

$$f(\mathbf{x}_m, \mathbf{u}_{m+1}) = \left({}^W_{k-1}T \right) \left({}^m_{m+1}T \right) \quad (4.36)$$

donde $\left({}^W_{k-1}T \right)$ es la transformación que representa la pose incierta de referencia \mathbf{x}_m con respecto al marco de referencia del mapa, y $\left({}^m_{m+1}T \right)$ representa la transformación preintegrada estimada por las mediciones inerciales

$$\left({}^m_{m+1}T \right) \sim \mathcal{N} \left([\boldsymbol{\omega} \cdot \delta t], \mathbf{I}^3 \eta \cdot \delta t^2 \right). \quad (4.37)$$

Sin embargo, debido a la formulación compacta de la función de costo para la optimización, ya no es posible derivar un estimador para la covarianza marginal de \mathbf{x}_{m+1} de una manera análoga a la derivada para el caso de Bundle Adjustment. Por lo tanto, se utiliza una propagación de primer orden según lo presentado por Eudes y Lhuillier [50].

4.4. Covarianza inicial de un punto

La triangulación estéreo es el proceso mediante el cual dos proyecciones visuales $\check{\mathbf{u}}_L \in \mathbb{R}^2$ y $\check{\mathbf{u}}_R \in \mathbb{R}^2$ que se detectan en las imágenes de la cámara izquierda y derecha respectivamente, y que se supone corresponden a una misma marca de referencia del ambiente, se pueden usar para reconstruir la posición ${}^W\mathbf{1}$ de esta última en el espacio. Las características de la imagen, como cualquier otra medida de sensor, son propensas a errores de medición y conllevan un cierto grado de incertidumbre, que debe propagarse a la estimación de la posición de la marca de referencia, por

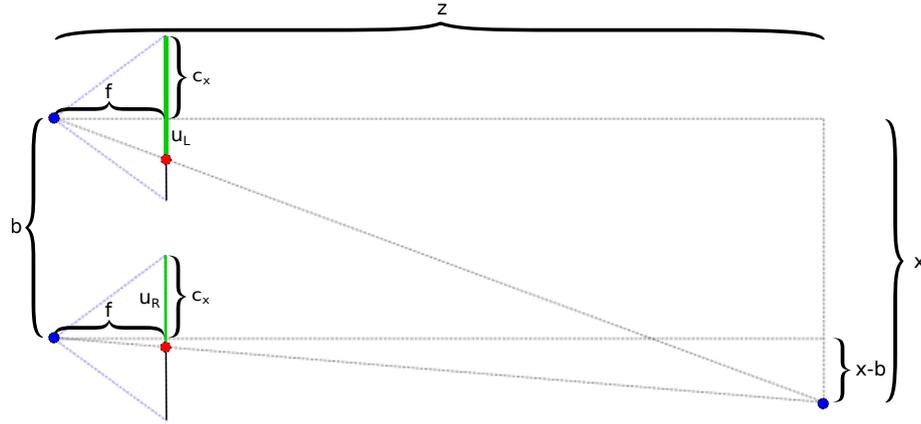


Figura 4.4: Triangulación estereó de un punto 3D en imágenes rectificadas.

lo tanto la posición es modelada como una v.a. siguiendo una distribución normal multivariada

$$w_{\mathbf{1}} \sim \mathcal{N}\left(\left[w_x \ w_y \ w_z \right]^T, w_{\Sigma_{\mathbf{1}}}\right). \quad (4.38)$$

Cuando se trabaja con cámaras rectificadas, como en el caso del presente trabajo, las matemáticas de la triangulación de características estereó probabilísticas se vuelven fácilmente manejables [75]. Dado que ambas mediciones se encuentran en el plano epipolar, comparten la misma coordenada vertical. Sean $\check{\mathbf{u}}_L = \begin{bmatrix} u_L & v \end{bmatrix}$ y $\check{\mathbf{u}}_R = \begin{bmatrix} u_R & v \end{bmatrix}$ características que observan una medición en la posición ${}^C\mathbf{X}$. La medición se puede reescribir como $\check{\mathbf{z}} = \begin{bmatrix} u_L & v & u_R \end{bmatrix}$. Analizando la Figura 4.4 se derivan las siguientes relaciones:

$$\frac{c_x}{c_z} = \frac{u_L - c_x}{f} \quad (4.39)$$

$$\frac{c_x - b}{c_z} = \frac{u_R - c_x}{f} \quad (4.40)$$

y consecuentemente

$$\frac{c_z}{f} = \frac{c_x}{u_L - c_x} = \frac{c_x - b}{u_R - c_x}. \quad (4.41)$$

Luego la función de triangulación se puede definir como

$$f(\check{\mathbf{z}}) = \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} = \begin{bmatrix} (u_L - c_x) \frac{b}{(u_L - u_R)} \\ (v - c_y) \frac{b}{(u_L - u_R)} \\ \frac{fb}{(u_L - u_R)} \end{bmatrix}. \quad (4.42)$$

Dado que f es no lineal, la propagación de la covarianza se puede aproximar mediante una linealización de primer orden

$${}^C\Sigma_{\mathbf{X}} = J_f \Sigma_{\mathbf{z}} J_f^\top, \quad (4.43)$$

donde J_f es la derivada de primer orden de f , es decir

$$\frac{\partial f}{\partial \check{\mathbf{z}}} = \begin{bmatrix} \frac{\partial f}{\partial u_L} & \frac{\partial f}{\partial v} & \frac{\partial f}{\partial u_R} \end{bmatrix} = \begin{bmatrix} b/d - b(u_L - c_x)/d^2 & 0 & b(u_L - c_x)/d^2 \\ -b(v - c_y)/d^2 & b/d & b(v - c_y)/d^2 \\ -fb/d^2 & 0 & fb/d^2 \end{bmatrix}, \quad (4.44)$$

o en términos de la posición de la marca de referencia

$$\frac{\partial f}{\partial \check{\mathbf{z}}} = \begin{bmatrix} \frac{\partial f}{\partial u_L} & \frac{\partial f}{\partial v} & \frac{\partial f}{\partial u_R} \end{bmatrix} = \begin{bmatrix} b/d - c_x/d & 0 & c_x/d \\ -c_y/d & b/d & c_y/d \\ -c_z/d & 0 & c_z/d \end{bmatrix}, \quad (4.45)$$

usando d para la disparidad $d = (u_L - u_R)$. La incertidumbre de las coordenadas de la imagen se puede tratar como no correlacionada con una variación de un píxel tomando $\Sigma_{\mathbf{z}} = \mathbf{I}$.

Esto produce una estimación distribuida normalmente para la posición de las marcas de referencia en las coordenadas de la cámara izquierda, pero en el marco actual debe representarse globalmente en las coordenadas del mundo. Dado que se conoce la pose incierta de la cámara del cuadro \mathbf{C} en coordenadas del mundo, esto se puede lograr aplicando una transformación incierta como se explica en la Sección 3.4, enfoque adoptado también por Paton et al. [76].

4.5. Detección de valores atípicos

Otro modelo probabilístico empleado en el sistema actual es el de detectar valores atípicos de medición. Es un problema conocido y recurrente que al establecer asociaciones entre características locales de imagen y marcas de referencia existentes mediante comparaciones de descriptores, se pueden establecer correspondencias espurias que generan mediciones que no se corresponden con la realidad. Sin embargo, el sistema se basa en el hecho de que estas coincidencias espurias son mucho menos numerosas que las buenas, por lo que no obstaculizan la optimización para llegar a una solución decente.

Intuitivamente, después de realizar una optimización exitosa, los errores de medición asociados a cada una de las características espurias deben ser significativamente mayores que los relacionados con las buenas mediciones. Esto se debe al hecho de que la optimización reduce cuidadosamente la influencia de los residuos grandes sobre los más pequeños mediante la aplicación de una función de costo robusta. Sin embargo, es útil detectar estas coincidencias espurias para reducir el ruido de

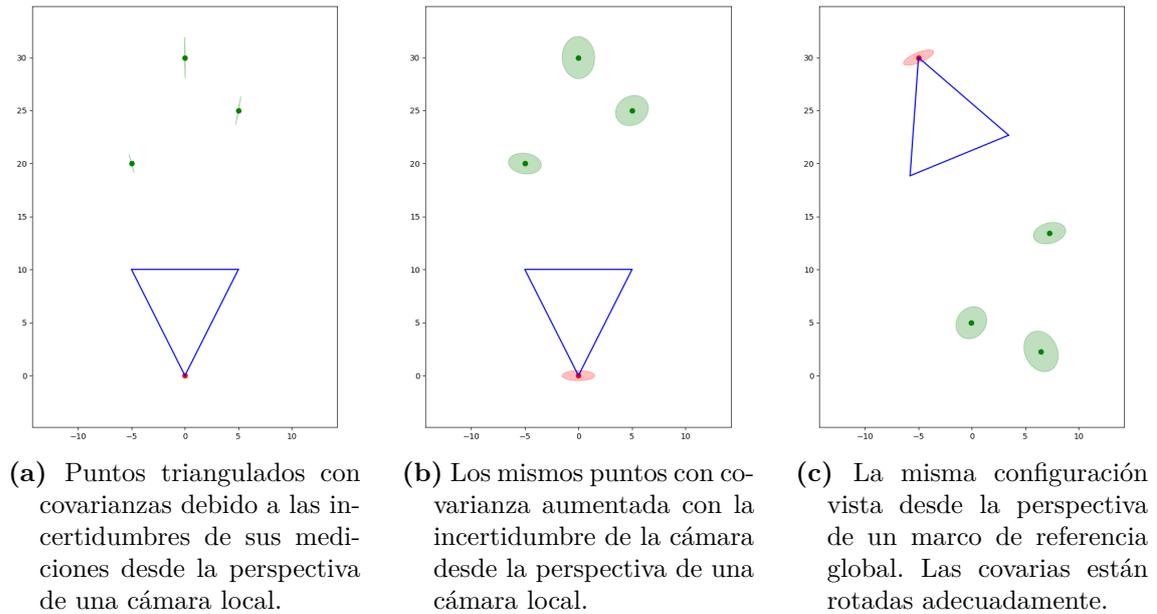


Figura 4.5: Cambio de referencia progresivo de la covarianza de una marca de referencia del ambiente.

optimización en el futuro y ya que pueden estar insinuando parámetros que no están lo suficientemente sólidos como para ser rastreados desde diferentes puntos de vista.

Por lo tanto, resulta útil examinar más de cerca las distribuciones de error de los residuos de optimización y definir un umbral a partir del cual se puede sospechar que una medición es un valor atípico. Esto se puede hacer considerando que las mediciones de posición de marcas de referencia del ambiente tienen una distribución de error gaussiana multivariada. Por lo tanto, cualquier proyección de marcas de referencia que coincida con esa característica en particular, considerando que la posición de la cámara y la posición de la marca de referencia se estiman con suficiente certeza, debe estar dentro de un intervalo de confianza significativo de la función de distribución de la característica.

Recordando la Sección 3.2.3 y la suposición mencionada anteriormente de que la incertidumbre en las coordenadas de la imagen característica se modela como no correlacionada con una varianza de un píxel, una medición se marca como un valor atípico si después de la optimización la relación

$$\chi_{k,\alpha}^2 < \mathbf{r}^T \mathbf{r} \quad (4.46)$$

se mantiene, donde $\mathbf{r} \in \mathbb{R}^k$ representa el vector de error medio. En el presente caso, se utiliza un intervalo de confianza de $\alpha = 0,95$ para rechazar valores atípicos para mediciones monoculares ($k = 2$) y estéreo ($k = 4$), que arrojan valores de umbral

de $\chi^2_{2,0,95} = 5,991$ y $\chi^2_{4,0,95} = 9,488$ respectivamente.

5 Bundle Adjustment local en tiempo constante

El capítulo anterior introduce el concepto de Bundle Adjustment y cómo se vuelve esencial para mantener el mapa en un estado óptimo y consistente para su uso futuro. También queda claro que realizar el Bundle Adjustment global (GBA) tiene una complejidad computacional que escala con el tamaño potencialmente ilimitado del mapa. Por lo tanto, cuando las aplicaciones objetivo deben responder en tiempo real, como es el caso de la navegación autónoma en robots móviles, el costo de computar la solución completa se vuelve prohibitivo a largo plazo.

El concepto de Bundle Adjustment local (LBA) [49] apunta a superar esta limitación optimizando solo una ventana de tamaño fijo o una parte relevante del mapa, típicamente relacionada con la vecindad de la pose actual. Se hace una fuerte suposición al afirmar que las estimaciones de parámetros más antiguos han convergido, o que no están lo suficientemente relacionadas con las mediciones recientemente incorporadas como para verse afectadas al considerarse estas últimas. Este enfoque plantea dos cuestiones importantes que deben abordarse: cómo debe seleccionarse, dada una restricción en el número de parámetros, la ventana local de manera que capture la información más relevante, y qué se debe hacer con los datos que se han omitido para mantener la consistencia global de los estimadores probabilísticos.

En este capítulo, se proponen dos estrategias para superar la complejidad ilimitada de GBA al realizar la optimización solo en una parte relevante y de tamaño acotada del mapa. Primero se introduce una estrategia de selección de ventana que tiene como objetivo seleccionar las variables que comparten la mayor cantidad de información con los datos recientemente incorporados. Posteriormente, se propone un modelo para mantener la consistencia global del conjunto de variables estimadas sin la necesidad de considerar la totalidad del mapa.

Se hará una suposición importante para simplificar el problema, al establecer que la cantidad de marcas de referencia que son visibles desde un conjunto de fotogramas clave en una región está delimitada por una constante. Esto es bastante razonable teniendo en cuenta que la cantidad de características que pueden detectarse en una imagen está limitada por el tamaño de la misma y, además, que los detectores de características más comunes producen conjuntos de puntos bastante esparsos. Por lo tanto, las políticas de ventana y sus complejidades asociadas se definirán solo en términos del número de fotogramas clave involucrados, y se asumirá implícitamente que cada vez que se inserte un fotograma clave en una ventana, también se

seleccionarán todas las marcas de referencia relacionadas.

5.1. Ventana local de parámetros

La rutina de optimización de mapa del sistema SLAM se encarga de refinar periódicamente el conjunto de variables estimadas a medida que se insertan nuevos parámetros y restricciones en el mapa. Sin embargo, a medida que el mapa crece, más y más variables sufrirán poco o ningún cambio significativo después de cada una de estas optimizaciones. Específicamente, estos serán los parámetros que comparten poca información con los datos recientemente incorporados. Dado que las operaciones de optimización son costosas y dependen en gran medida del número de parámetros calculados, se vuelve deseable detectar las partes más afectadas del mapa y optimizar solo este conjunto reducido de variables.

El problema de elegir una ventana de optimización se puede definir entonces como una selección, dado un conjunto de fotogramas clave de consulta (que normalmente corresponden a datos incorporados recientemente), el conjunto de variables y relaciones que se verán significativamente afectadas por las nuevas restricciones durante una (hipotética) optimización global. De manera intuitiva, estas serán las variables que se encuentran en una vecindad espacial de los fotogramas clave de consulta. Además el conjunto de estas variables debe poder ser seleccionado de manera tal que no dependa del número total de variables en el mapa. Por este motivo, se implementa aquí una estructura de datos personalizada similar a un grafo, comúnmente conocida como grafo de co-visibilidad.

Un grafo de co-visibilidad se puede considerar como una aumentación del típico grafo de mediciones de marcas del ambiente por poses de sensores, donde se agregan restricciones entre las poses que codifican el peso de co-visibilidad, es decir, el número de marcas de referencia detectados por ambas (ver Figura 5.1). Esto proporciona una métrica razonable para el grado en que dos fotogramas clave están relacionados entre sí en un mapa. Por lo tanto, se podría construir una ventana de optimización seleccionando incrementalmente, a partir de los fotogramas clave de consulta, los fotogramas clave con el mayor peso de co-visibilidad respecto de algún fotograma clave que ya esté en la ventana, hasta llegar a un número límite de parámetros a optimizar. Este es, de hecho, un enfoque común en la literatura de SLAM [3, 77], y será el método de selección de ventana de elección en el trabajo actual.

Sin embargo, se debe tener cuidado al omitir los parámetros de la optimización que mantienen restricciones directas a otros dentro de la ventana seleccionada. Dado que estas restricciones quedarán excluidas de la optimización, algunos elementos dentro de la ventana pueden dejarse más limitados de lo que serían si se considerara todo el problema canónico. Por lo tanto, pueden divergir de la solución deseada, afectando aún más las estimaciones de los elementos restantes en la ventana. También puede surgir un caso extremo en el que dichos elementos se vuelvan insuficientemente

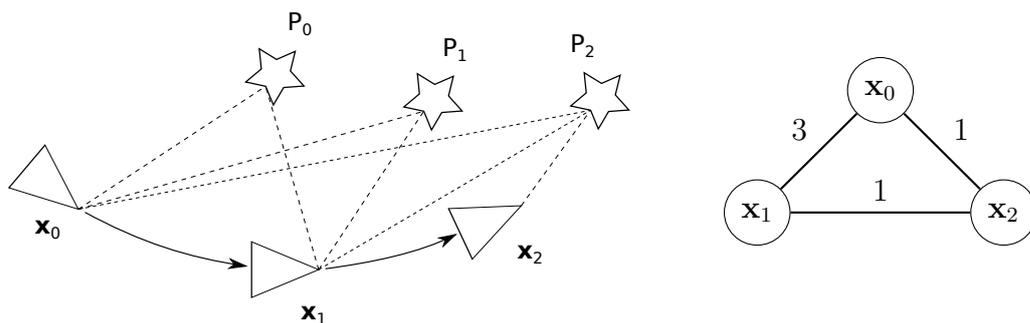


Figura 5.1: Una instancia de un problema simple de SLAM (izquierda) y el grafo de co-visibilidad correspondiente (derecha).

restringidos, por lo que pueden quedar algunos parámetros (al menos parcialmente) indeterminados.

Como se mencionó anteriormente, el método de selección de ventana propuesto en el trabajo actual se enfoca exclusivamente en los fotogramas clave. Una vez que se ha seleccionado un conjunto de fotogramas clave, todas las marcas de referencia relacionadas con cada uno de esos fotogramas clave también se incorporan en la ventana local. Sin embargo, algunos fotogramas clave relacionados con estas marcas pueden haberse omitido. Por lo tanto, una parametrización completa de las mismas puede verse comprometida, por ejemplo, seleccionando una marca de referencia (3 grados de libertad) que solo está relacionada por una medición monocular (2 grados de libertad) a un determinado fotograma clave, a pesar de que existían mediciones adicionales de los fotogramas clave como parte de la solución canónica. En la práctica, estos casos se pueden detectar fácilmente y se dejan fuera de la ventana de optimización, y dado que los casos son raros y las poses de fotogramas clave suelen estar muy sobreparametrizadas, es raro, aunque no imposible, que un fotograma clave quede (al menos parcialmente) indeterminado. En la práctica, como medida de seguridad, este caso se verifica pero no se tiene en cuenta una solución.

5.2. Eliminación de parámetros

El estimador máximo a posteriori para el problema de Bundle Adjustment presentado en Sección 4.3 se vuelve imposible de computar a largo plazo a medida que crece el número de poses y marcas de referencia del mapa. Por lo tanto, resulta conveniente encontrar una manera de eliminar los parámetros del problema de optimización original sin perder demasiada información. Una vez que se ha seleccionado un con-

junto de fotogramas clave para la optimización, se debe tener cuidado en cuanto a cómo se manejan los parámetros omitidos para preservar la consistencia probabilística con respecto a la solución canónica. Desde un punto de vista probabilístico, se pueden mencionar dos enfoques estándar para la eliminación de parámetros: condicionamiento y marginalización.

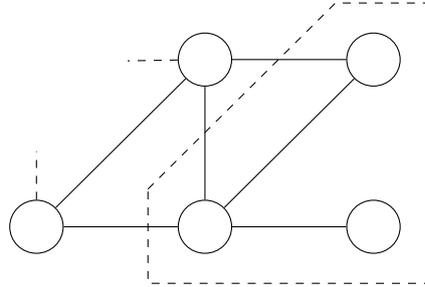


Figura 5.2: Parte de una red bayesiana que representa una densidad conjunta. Las variables dentro del área punteada representan la ventana de optimización seleccionada. Por simplicidad, solo se representan las variables latentes, mientras que las variables observadas están modeladas implícitamente por las aristas.

El condicionamiento asume que los parámetros lejanos ya han convergido a sus valores reales. En tal caso, no habría ninguna duda sobre su verdadera naturaleza y no tendría sentido seguir tratándolos como variables aleatorias probabilísticas, por lo que son consideradas como constantes. Alternativamente, se puede pensar que las variables condicionadas son promovidas a tener información infinita. En cualquier caso, esto conduce a una estimación altamente confiada, y por lo tanto no es adecuado para realizar SLAM que sea probabilísticamente consistente globalmente. Sin embargo, si el cálculo correcto de la covarianza fuera de poca importancia, el acondicionamiento es bastante sencillo de realizar en tiempo constante en el problema de Bundle Adjustment simplemente descartando las mediciones omitidas, y sigue siendo una aproximación razonable [59] para encontrar las estimaciones de las medias del problema original.

Por otro lado, la marginalización es el proceso mediante el cual se integra sobre las variables excluidas, por lo que aún se consideran todos sus estados posibles, aunque no están explícitamente modelados. Por lo tanto, la optimización sobre un problema marginalizado produce exactamente la misma distribución para el subconjunto de variables seleccionadas como si se hubieran computado mediante la optimización global. Esta es una gran diferencia respecto del condicionamiento de variables, donde la distribución calculada sería bastante diferente.

En particular para distribuciones normales multivariadas, el cálculo de la información marginal es posible realizando operaciones algebraicas sobre los parámetros variables (ver Sección 3.2.1), produciendo un conjunto de parámetros para la distribución marginal que también resulta tener distribución normal. Si bien la complejidad de calcular la verdadera marginal del problema en cada paso es una función que

depende del número total de variables, la actualización de la información marginal es posible en un tiempo constante al agregar elementos recientemente marginalizados a la marginal del paso anterior.

5.2.1. Marginalizando parámetros en ecuaciones de cuadrados mínimos

Considere el problema de estimar los parámetros de un conjunto de variables $X \sim \mathcal{N}(\mathbf{x}, \Sigma)$ que siguen una distribución normal multivariada optimizando sobre un conjunto de mediciones normalmente distribuidas. Recordando la ecuación 4.22 esto puede lograrse mediante la estimación MAP aplicando una optimización lineal de cuadrados mínimos sobre un conjunto de ecuaciones normales con la estructura

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \quad (5.1)$$

Podría darse el caso donde solo un subconjunto arbitrario $X_b \subseteq X$ de estas variables sea considerado de interés, y calcular una solución para X puede ser prohibitivamente costoso. En tal caso, podría resultar conveniente estimar solo la distribución marginal de los parámetros interesantes X_b .

Dada la partición $X = X_a \cup X_b$ y siguiendo la ecuación 3.5, la densidad conjunta estimada de los parámetros calculados por la solución completa también se puede dividir como

$$[X_a, X_b] \sim \mathcal{N} \left(\begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix}, \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ab}^T & \Sigma_{bb} \end{bmatrix} \right). \quad (5.2)$$

Recordando la ecuación 3.8, una optimización adecuada de la distribución marginal de los parámetros X_b debería producir un estimador para

$$X_b \sim \mathcal{N}(\mathbf{x}_b, \Sigma_{bb}). \quad (5.3)$$

El enfoque habitual [78, 79, 80, 81, 56] para la marginalización de parámetros en los problemas de Bundle Adjustment es manipular directamente las ecuaciones de cuadrados mínimos para lograr el resultado deseado.

La piedra angular de este enfoque algebraico lineal para la marginalización es el hecho de que, dada una ecuación lineal de mínimos cuadrados $\mathbf{A}\mathbf{x} = \mathbf{b}$ con la siguiente estructura de bloque

$$\begin{bmatrix} \mathbf{A}_a & \mathbf{A}_b \\ \mathbf{0} & \mathbf{A}_c \end{bmatrix} \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix} = \begin{bmatrix} \mathbf{b}_a \\ \mathbf{b}_b \end{bmatrix}, \quad (5.4)$$

si un valor óptimo $\mathbf{x} = \begin{bmatrix} \mathbf{x}_a & \mathbf{x}_b \end{bmatrix}$ existe, también se puede encontrar \mathbf{x}_b resolviendo

la ecuación de mínimos cuadrados más pequeña $\mathbf{A}_c \mathbf{x}_b = \mathbf{b}_b$.

Considere cualquier ecuación lineal de mínimos cuadrados $\mathbf{A} \mathbf{x} = \mathbf{b}$ que estima una densidad normal multivariada y se puede dividir de acuerdo a la ecuación 5.2. La ecuación lineal puede, bajo el reordenamiento adecuado de las variables, particionarse siguiendo la siguiente estructura de bloque

$$\begin{bmatrix} \mathbf{A}_{aa} & \mathbf{A}_{ab} \\ \mathbf{A}_{ab}^T & \mathbf{A}_{bb} \end{bmatrix} \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix} = \begin{bmatrix} \mathbf{b}_a \\ \mathbf{b}_b \end{bmatrix}. \quad (5.5)$$

Por lo tanto, se busca una matriz \mathbf{C} tal que $\mathbf{C} \mathbf{A}$ produce una matriz con la estructura de bloque mostrada en la ecuación 5.4. Se debe tener en cuenta que resolver $\mathbf{A} \mathbf{x} = \mathbf{b}$ para \mathbf{x} es lo mismo que resolver $\mathbf{C} \mathbf{A} \mathbf{x} = \mathbf{C} \mathbf{b}$, es decir

$$\begin{bmatrix} \mathbf{C}_a & \mathbf{C}_b \\ \mathbf{C}_c & \mathbf{C}_d \end{bmatrix} \begin{bmatrix} \mathbf{A}_{aa} & \mathbf{A}_{ab} \\ \mathbf{A}_{ab}^T & \mathbf{A}_{bb} \end{bmatrix} \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix} = \begin{bmatrix} \mathbf{C}_a & \mathbf{C}_b \\ \mathbf{C}_c & \mathbf{C}_d \end{bmatrix} \begin{bmatrix} \mathbf{b}_a \\ \mathbf{b}_b \end{bmatrix} \quad (5.6)$$

o, distribuyendo el producto,

$$\begin{bmatrix} \mathbf{C}_a \mathbf{A}_{aa} + \mathbf{C}_b \mathbf{A}_{ab}^T & \mathbf{C}_a \mathbf{A}_{ab} + \mathbf{C}_b \mathbf{A}_{bb} \\ \mathbf{C}_c \mathbf{A}_{aa} + \mathbf{C}_d \mathbf{A}_{ab}^T & \mathbf{C}_c \mathbf{A}_{ab} + \mathbf{C}_d \mathbf{A}_{bb} \end{bmatrix} \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix} = \begin{bmatrix} \mathbf{C}_a \mathbf{b}_a + \mathbf{C}_b \mathbf{b}_b \\ \mathbf{C}_c \mathbf{b}_a + \mathbf{C}_d \mathbf{b}_b \end{bmatrix}. \quad (5.7)$$

Recordando la estructura objetivo de la matriz (ecuación 5.4) se deduce que $\mathbf{C}_c \mathbf{A}_{aa} + \mathbf{C}_d \mathbf{A}_{ab}^T = \mathbf{0}$, luego encontrar una solución para \mathbf{x}_b es equivalente a resolver

$$(\mathbf{C}_c \mathbf{A}_{ab} + \mathbf{C}_d \mathbf{A}_{bb}) \mathbf{x}_b = \mathbf{C}_c \mathbf{b}_a + \mathbf{C}_d \mathbf{b}_b. \quad (5.8)$$

Es interesante observar que solo son importantes \mathbf{C}_c y \mathbf{C}_d (siempre que no se anulen filas por la pre-multiplicación de \mathbf{C}). Una solución posibles es

$$\mathbf{C} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{A}_{ab}^T \mathbf{A}_{aa}^{-1} & \mathbf{I} \end{bmatrix}, \quad (5.9)$$

que insertada en la ecuación 5.8 presenta una ecuación de mínimos cuadrados para estimar la distribución marginal de X_b

$$(\mathbf{A}_{bb} - \mathbf{A}_{ab}^T \mathbf{A}_{aa}^{-1} \mathbf{A}_{ab}) \mathbf{x}_b = \mathbf{b}_b - \mathbf{A}_{ab}^T \mathbf{A}_{aa}^{-1} \mathbf{b}_a. \quad (5.10)$$

El término $\mathbf{A}_{bb} - \mathbf{A}_{ab}^T \mathbf{A}_{aa}^{-1} \mathbf{A}_{ab}$ es llamado el *Complemento de Schur* del bloque \mathbf{A}_{aa} en \mathbf{A} , y se nota $\mathbf{A}/\mathbf{A}_{aa}$, por lo tanto

$$(\mathbf{A}/\mathbf{A}_{aa}) \mathbf{x}_b = \mathbf{b}_b - \mathbf{A}_{ab}^T \mathbf{A}_{aa}^{-1} \mathbf{b}_a. \quad (5.11)$$

Recordando la Sección 4.3.1.3 se deduce que la matriz de información para la dis-

tribución marginal de X_b es $\mathbf{A}/\mathbf{A}_{aa}$, luego

$$X_b \sim \mathcal{N}(\mathbf{x}_b, (\mathbf{A}/\mathbf{A}_{aa})^{-1}). \quad (5.12)$$

Si bien las variables marginalizadas no se modelan explícitamente en el problema de optimización resultante, la solución calculada refleja el estado de la solución canónica. Sin embargo, esto es estrictamente correcto solo bajo el supuesto de que las restricciones consideradas son lineales. En el caso de mínimos cuadrados no lineales, las restricciones aproximan procesos no lineales mediante ecuaciones de Taylor de primer orden, y el punto de linealización se refina después de cada iteración aprovechando los estimadores optimizados. Debido a que las variables marginalizadas pueden optimizarse implícitamente, y que las mediciones asociadas no se pueden volver a re-linealizar, se producen pequeños errores. Estos errores de linealización se propagarán luego al resto del problema, incluidas las variables dentro de la ventana. Por otra parte, estas variables generalmente se omiten suponiendo que no se verán afectadas significativamente por la nueva información y las optimizaciones, por lo que estos errores generalmente se ignoran.

5.2.2. Marginalizar parámetros en problemas ralos

En el caso particular del Bundle Adjustment de un conjunto de parámetros (potencialmente multivariados) $X = \{x_i\}$ relacionados por un conjunto de restricciones $Z = \{f_k(X_k)\}$, donde cada restricción f_k a su vez se relaciona con un subconjunto arbitrario de parámetros $X_k \subseteq X$, la matriz del sistema de mínimos cuadrados es la matriz de información $\mathbf{\Lambda} = \mathbf{J}^T \mathbf{W}^{-1} \mathbf{J}$ para la distribución conjunta de \mathbf{X} , como lo muestra la ecuación 4.22. Cuanto más escaso es el conjunto de restricciones, más eficiente es calcular la matriz de información mediante el cálculo de bloques individuales de la matriz en términos del conjunto de relaciones. Además de ser explotado por el optimizador para acelerar la multiplicación de matrices, también se puede aprovechar la estructura rala de la matriz para realizar la marginalización, en particular calcular el complemento de Schur, de una manera más eficiente.

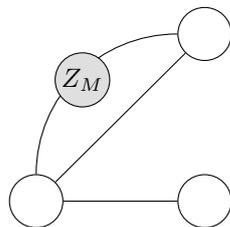


Figura 5.3: Red bayesiana que representa la ventana de optimización marginal introducida en la Figura 5.2. Una nueva medición abstracta Z_M captura la información de las variables marginadas como propagadas en las restantes.

Supóngase que, sin pérdida de generalidad (reordenar los parámetros en teoría no

altera los resultados), el conjunto de parámetros de izquierda / superior $\mathbf{x}_a \in X$ en la matriz hessiana $\mathbf{\Lambda}$ deben ser marginados utilizando el complemento Shur. Particionando los elementos del sistema como

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_a \\ \mathbf{b}_b \end{bmatrix} \quad \mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_{aa} & \mathbf{\Lambda}_{ab} \\ \mathbf{\Lambda}_{ab}^T & \mathbf{\Lambda}_{bb} \end{bmatrix} \quad (5.13)$$

la ecuación de mínimos cuadrados normal para la distribución marginal de \mathbf{x}_b (ecuación 5.10) puede escribirse como

$$(\mathbf{\Lambda}/\mathbf{\Lambda}_{aa}) \mathbf{x}_b = \mathbf{b}_b - \mathbf{\Lambda}_{ab}^T \mathbf{\Lambda}_{aa}^{-1} \mathbf{b}_a. \quad (5.14)$$

Recordando que \mathbf{J} representa el jacobiano de las mediciones

$$\mathbf{J} = \frac{\partial \mathbf{z}}{\partial \mathbf{x}} \quad (5.15)$$

y suponiendo, como es habitual en un contexto SLAM, que las mediciones son condicionalmente independientes entre sí ($\mathbf{W} = \text{diag}(\mathbf{W}_k)$), la matriz de información $\mathbf{\Lambda} = \mathbf{J}^T \mathbf{W}^{-1} \mathbf{J}$ se puede descomponer en bloques individuales correspondientes a cada par de parámetros \mathbf{x}_i y \mathbf{x}_j

$$\mathbf{\Lambda}_{ij} = \sum_k \frac{\partial f_k(X_k)^T}{\partial x_i} \mathbf{W}_k^{-1} \frac{\partial f_k(X_k)}{\partial x_j}. \quad (5.16)$$

Es sencillo observar que los términos de suma individuales son distintos de cero solo si $\{x_i, x_j\} \subset X_k$, es decir, si ambos parámetros están directamente relacionados a través de la restricción f_k .

Recordando la ecuación 5.10 la matriz marginal del sistema de mínimos cuadrados $\mathbf{\Lambda}/\mathbf{\Lambda}_{aa} = \mathbf{\Lambda}_{bb} - \mathbf{\Lambda}_{ab}^T \mathbf{\Lambda}_{aa}^{-1} \mathbf{\Lambda}_{ab}$ puede pensarse nuevamente en términos de los bloques de parámetros restantes, donde el factor $S = \mathbf{\Lambda}_{ab}^T \mathbf{\Lambda}_{aa}^{-1} \mathbf{\Lambda}_{ab}$ adopta una estructura en la que cada bloque representa nuevas relaciones inducidas sobre cada par de parámetros restantes \mathbf{x}_i y \mathbf{x}_j

$$S_{ij} = \mathbf{\Lambda}_{ai}^T \mathbf{\Lambda}_{aa}^{-1} \mathbf{\Lambda}_{aj}. \quad (5.17)$$

Siguiendo el mismo razonamiento que en la ecuación 5.16, S_{ij} no es cero solo si los parámetros \mathbf{x}_i y \mathbf{x}_j están directamente relacionados por alguna restricción a cualquiera de los elementos dejados fuera en X_a . Estos elementos conforman lo que se denominará la frontera de la ventana de parámetros. Una frontera se puede dividir en tres partes distintas: el conjunto de parámetros que pertenecen a la ventana que están directamente relacionados con los elementos que quedan fuera de la ventana, el conjunto análogo de los parámetros dejados fuera que están directamente relacionados con los parámetros dentro de la ventana, y el conjunto de restricciones que conectan estos parámetros.

Se puede hacer un ejemplo ilustrativo de la marginalización de parámetros al marginalizar un solo bloque de parámetros arbitrario \mathbf{x}_0 . La matriz de información marginal resultante $\mathbf{\Lambda}/\mathbf{\Lambda}_0$ para el problema reducido se compone de bloques individuales que relacionan cada par de parámetros restantes \mathbf{x}_i y \mathbf{x}_j ($i, j \neq 0$)

$$(\mathbf{\Lambda}/\mathbf{\Lambda}_0)_{ij} = \mathbf{\Lambda}_{ij} + \mathbf{\Lambda}_{0i}^T \mathbf{\Lambda}_{00}^{-1} \mathbf{\Lambda}_{0j}. \quad (5.18)$$

Aquí queda claro que la nueva matriz del sistema $\mathbf{\Lambda}/\mathbf{\Lambda}_0$ estará contaminada por términos marginales distintos a cero $\mathbf{\Lambda}_{0i}^T \mathbf{\Lambda}_{00}^{-1} \mathbf{\Lambda}_{0j}$ para cada par de parámetros \mathbf{x}_i y \mathbf{x}_j que estaban directamente relacionado por alguna restricción con el bloque de parámetro marginalizado \mathbf{x}_0 , incluso si no estaban directamente relacionados de antemano. Esto es bastante intuitivo, ya que la restricción implícita entre \mathbf{x}_i y \mathbf{x}_j que era impuesta por sus respectivas relaciones con \mathbf{x}_0 ahora se modela como una restricción explícita entre ellas.

En el caso general, la marginalización genera nueva información distinta de cero para cada par de parámetros restantes que tenían mediciones a alguna variables marginalizada. Esto produce relleno de la matriz de información para bloques de variables que no estaban originalmente relacionados, y por lo tanto una densificación de la matriz del sistema de mínimos cuadrados que aumenta la complejidad computacional del problema. Hay un intercambio a hacer aquí: la marginalización reduce el número de ecuaciones para un problema, pero a su vez produce un problema mucho más denso para ese nuevo conjunto reducido de ecuaciones. Ciertas heurísticas [57, 82, 83] pueden emplearse para garantizar un cierto nivel de esparsidad que mitiga este problema, pero el tema no se tratará en el presente trabajo. En teoría, dado que la matriz de mínimos cuadrados marginal está delimitada por un número constante de parámetros y la matriz de optimización del problema completos, aunque es escasa, depende del número de parámetros, el número de estos eventualmente superará dicha constante.

5.2.3. Actualizando la distribución marginal incrementalmente

Una propiedad útil de calcular distribuciones marginales utilizando el complemento de Schur es que las estimaciones previamente calculadas de distribuciones marginales pueden actualizarse incrementalmente con nuevos elementos, siempre que no estén directamente relacionadas con parámetros ya marginalizados. Es decir

$$\begin{bmatrix} \mathbf{\Lambda} & \cdots \\ \vdots & \mathbf{\Lambda}_B \end{bmatrix} / \mathbf{\Lambda}_{AA} = \begin{bmatrix} \mathbf{\Lambda}/\mathbf{\Lambda}_{AA} & \cdots \\ \vdots & \mathbf{\Lambda}_B \end{bmatrix} \quad (5.19)$$

donde $\mathbf{\Lambda}_{AA}$ es una submatriz de $\mathbf{\Lambda}$ y vale que $\mathbf{\Lambda}_{AB} = 0$. Otra propiedad útil es el hecho de que marginalizar parámetros adicionales de una distribución marginal con

el complemento de Schur es un proceso transparente, es decir,

$$\Lambda/\Lambda_{A \cup B} = (\Lambda/\Lambda_A)/\Lambda_B. \quad (5.20)$$

Estas propiedades resultan útiles ya que permiten realizar una estimaciones probabilísticamente consistentes de los parámetros globales al realizar un Bundle Adjustment local sin la necesidad de volver a calcular la distribución marginal total en cada paso. Cada vez que el problema crece, siempre que los nuevos parámetros no estén directamente relacionados con los que ya están marginalizados, el factor acumulado representa (hasta los errores de linealización) el verdadero marginal de los parámetros activos con respecto al problema global. Siempre que un nuevo parámetro sea marginalizado, puede hacerse actualizando el factor anterior con las mediciones relevantes. Al retener la información completa del problema, también es posible calcular y actualizar las distribuciones marginales de los parámetros individuales dentro de la ventana, después de cada optimización exitosa, en complejidad de tiempo constante.

Sin embargo, las restricciones introducidas por los sucesivos procedimientos de marginalización se vuelven indistinguibles entre sí, y también de las definidas en la declaración original del problema. Esto se convierte en un problema cuando se vuelven a visitar áreas previamente marginalizadas, una ocurrencia común en SLAM, ya que requiere la reincorporación de poses y medidas marginalizadas en el mapa local, lo que requeriría revertir el proceso de marginalización hasta la pose actual desde el momento a partir del cuál se cerró el ciclo. El tamaño de dicho ciclo, y por lo tanto la complejidad computacional, podría ser potencialmente ilimitado. Sin embargo, la verdadera marginal del problema sigue siendo un buen estándar contra el cual medir, y de la cual derivar, otras heurísticas. Para poder seguir realizando Bundle Adjustment local en un tiempo constante y que siga siendo probabilísticamente constante con respecto a la solución global, incluso en presencia de cierres de ciclo, una de estas heurísticas se propondrá en la siguiente sección: el *Prior Unario*.

5.2.4. Prior unario

Dado que las distribuciones marginales de los parámetros individuales se pueden actualizar periódicamente siempre que se ajusten después de una optimización exitosa, los datos deberían estar disponibles en tiempo constante. Cualquier distribución marginal adecuada de un parámetro codifica la información de toda la densidad conjunta del problema original tal como la percibe. Por lo tanto, este trabajo propone aprovechar la distribución marginal de una de las variables seleccionadas para la optimización, la cuál será llamada *parámetro de referencia* de la ventana, como una manera de aproximar la información agregada sobre los parámetros restantes que se introducirían mediante una marginalización verdadera. Las figuras 5.3 y 5.4 ilustran la estructura de dos redes bayesianas que representan las estructuras de una ventana

correctamente marginalizada y la de la heurística propuesta, respectivamente, para el grafo original presentado en la Figura 5.2.

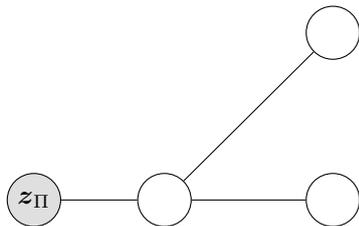


Figura 5.4: Red bayesiana que representa una aproximación de la ventana de optimización marginalizada representada en la Figura 5.3 usando un Prior Unario, modelado como una medición z_{Π} .

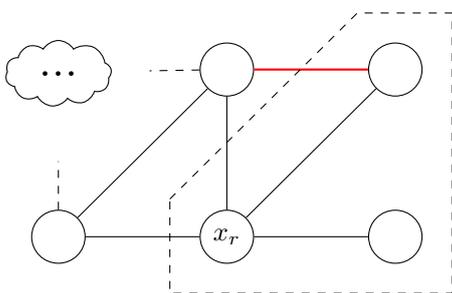


Figura 5.5: Representación como grafo de factores para un problema de SLAM. Una ventana está definida por la línea punteada junto al marco de referencia x_r .

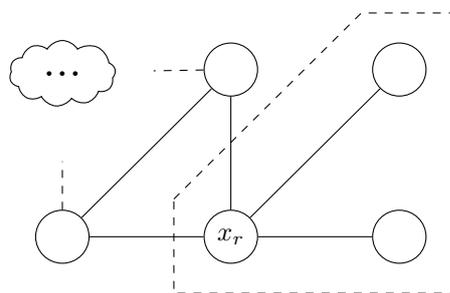


Figura 5.6: Reducción del problema canónico, donde simplemente se eliminan los bordes de la frontera que no están relacionados con el parámetro de referencia.

La heurística propuesta se definirá calculando la verdadera marginal de un subconjunto particular del problema original. Considerese, sin pérdida de generalidad, un problema de Bundle Adjustment con un conjunto de parámetros $X = \{\mathbf{x}_0 \dots \mathbf{x}_n\}$ y restricciones Z . Una ventana de parámetros $X_W = \{\mathbf{x}_0 \dots \mathbf{x}_m\}$, $X_w \subseteq X$, y mediciones relevantes $Z_W = \{f_k(X_k)\}$, $Z_W \subseteq Z$, se seleccionan para optimización, donde $\mathbf{x}_0 \in X$ es el parámetro de referencia mencionado para la ventana. Además, se supone que está disponible una distribución marginal $\mathbf{x}_0 \sim \mathcal{N}(\check{\mathbf{x}}_0, \check{\Sigma}_0)$ del parámetro de referencia con respecto al problema global.

El problema del subconjunto se definirá eliminando del problema original todas las restricciones de frontera que relacionan los parámetros de $X_W - \{\mathbf{x}_0\}$ a los parámetros fuera de la ventana. Es decir, la única conexión entre la ventana y el resto del problema seguirán siendo las restricciones que se relacionan directamente con \mathbf{x}_0 para cualquier parámetro fuera de la ventana. Como ejemplo, considere el grafo de factores para un problema como se muestra en la Figura 5.5, y el sub-grafo

relacionado que se muestra en la Figura 5.6 donde se eliminan todas las restricciones de frontera, excepto aquellas relacionado con el parámetro de referencia x_r .

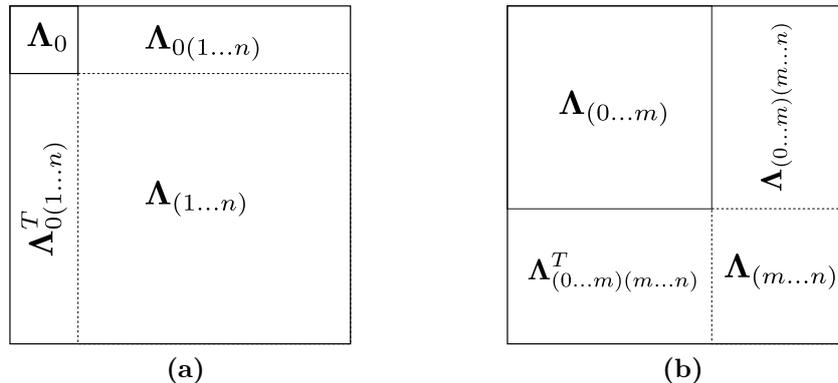


Figura 5.7: Dos particiones distintas de la misma matriz de información Λ .

Considere las particiones de la matriz de información Λ para el problema original representado en la Figura 5.7, y las particiones análogas para la matriz de información Γ del subproblema. Según la ecuación 5.16 la estructura de la matriz de información para el problema original se puede escribir como una suma de términos de factores

$$\Lambda = \sum_k \frac{\partial f_k(X_k)^T}{\partial x} \mathbf{W}_k^{-1} \frac{\partial f_k(X_k)}{\partial x}. \tag{5.21}$$

Esta suma se puede dividir a su vez en dos términos independientes, uno de los cuales conforma la matriz de información Γ para el subproblema mencionado y el complemento Δ conformado por los términos que quedaron fuera con respecto al problema original, es decir

$$\Lambda = \Gamma + \Delta. \tag{5.22}$$

Dado que las tres matrices son semidefinidas positivas por construcción, vale [84] que

$$\det(\Gamma) + \det(\Delta) \leq \det(\Lambda) \tag{5.23}$$

y luego

$$\det(\Gamma) \leq \det(\Lambda). \tag{5.24}$$

Por lo tanto, el subproblema menos informativo representa una solución conservadora con respecto al problema original.

Como esto también se aplica a las distribuciones marginales, la información marginal $\Gamma/\Gamma_{(m..n)(m..n)}$ se usará para aproximar la verdadera información marginal de

la ventana $\Lambda/\Lambda_{(m..n)(m..n)}$ con respecto a la declaración del problema original. Como se mostrará, se puede definir un algoritmo de tiempo constante para calcular $\Gamma/\Gamma_{(m..n)(m..n)}$ usando la distribución marginal disponible del parámetro de referencia \mathbf{x}_0 .

Considere la información marginal de los parámetros de la ventana X_W en el subproblema utilizando el complemento de Schur

$$\Gamma/\Gamma_{(m..n)(m..n)} := \Gamma_{(0..m)(0..m)} - \Gamma_{(m..n)(0..m)}^T \Gamma_{(m..n)(m..n)}^{-1} \Gamma_{(m..n)(0..m)}. \quad (5.25)$$

Al particionar los parámetros de la ventana X_W en un parámetro de referencia \mathbf{x}_0 y el resto de la ventana, por otro lado, la ecuación mencionada anteriormente puede reescribirse en términos de los bloques individuales relacionados

$$\Gamma/\Gamma_{(m..n)(m..n)} = \Gamma_{(0..m)(0..m)} - \begin{bmatrix} \Gamma_{(m..n)0}^T \\ \Gamma_{(m..n)(1..m)}^T \end{bmatrix} \Gamma_{(m..n)(m..n)}^{-1} \begin{bmatrix} \Gamma_{(m..n)0} & \Gamma_{(m..n)(1..m)} \end{bmatrix}, \quad (5.26)$$

donde por definición del subproblema, ya que ninguno de los parámetros en $X_W - \{\mathbf{x}_0\}$ se relaciona con los parámetros de izquierda, $\Gamma_{(m..n)(1..m)} = \mathbf{0}$. La información marginal de la ventana con respecto al subproblema por lo tanto se vuelve

$$\Gamma/\Gamma_{(m..n)(m..n)} = \Gamma_{(0..m)(0..m)} - \begin{bmatrix} \Gamma_{(m..n)0}^T \Gamma_{(m..n)(m..n)}^{-1} \Gamma_{(m..n)0} & 0 \\ 0 & 0 \end{bmatrix}, \quad (5.27)$$

donde la submatriz de información $\Gamma_{(0..m)(0..m)}$ puede ser computada como

$$\Gamma_{(0..m)(0..m)} = \sum_k \frac{\partial f_k(X_k)}{\partial [\mathbf{x}_0 \dots \mathbf{x}_m]}^T \mathbf{W}_k^{-1} \frac{\partial f_k(X_k)}{\partial [\mathbf{x}_0 \dots \mathbf{x}_m]}. \quad (5.28)$$

Se vuelve claro que los únicos términos que no son cero son aquellos en los que f_k implica al menos un parámetro de la ventana. Dado que se supone que ambas de estas cantidades están limitadas, el término se puede calcular en tiempo constante. Sin embargo, el segundo término todavía no resulta sencillo de computar en complejidad de tiempo constante, ya que requiere sub-matrices de información de cómputo que involucran a los parámetros omitidos.

Por otro lado, considere dos cantidades adicionales: la información marginal del parámetro de referencia

$$\Gamma/\Gamma_{(1..n)(1..n)} := \Gamma_{00} - \Gamma_{(1..n)0}^T \Gamma_{(1..n)(1..n)}^{-1} \Gamma_{(1..n)0}, \quad (5.29)$$

y la información marginal del parámetro de referencia con respecto a la sub-matriz de información de la ventana

$$\Gamma_{(0..m)(0..m)}/\Gamma_{(1..m)(1..m)} := \Gamma_{00} - \Gamma_{(1..m)0}^T \Gamma_{(1..m)(1..m)}^{-1} \Gamma_{(1..m)0}. \quad (5.30)$$

El primero se puede reescribir dividiendo nuevamente los parámetros de la ventana X_W en un parámetro de referencia \mathbf{x}_0 y el resto de la ventana

$$\mathbf{\Gamma}/\mathbf{\Gamma}_{(1..n)(1..n)} = \mathbf{\Gamma}_{00} - \begin{bmatrix} \mathbf{\Gamma}_{(1..m)0}^T & \mathbf{\Gamma}_{(m..n)0}^T \end{bmatrix} \begin{bmatrix} \mathbf{\Gamma}_{(1..m)(1..m)} & \mathbf{\Gamma}_{(1..m)(m..n)} \\ \mathbf{\Gamma}_{(1..m)(m..n)}^T & \mathbf{\Gamma}_{(m..n)(m..n)} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Gamma}_{(1..m)0} \\ \mathbf{\Gamma}_{(m..n)0} \end{bmatrix} \quad (5.31)$$

donde nuevamente $\mathbf{\Gamma}_{(1..m)(m..n)} = 0$, y dado que la inversa de una matriz diagonal en bloques es la matriz de bloques invertidos

$$\mathbf{\Gamma}/\mathbf{\Gamma}_{(1..n)(1..n)} = \mathbf{\Gamma}_{00} - \mathbf{\Gamma}_{(1..m)0}^T \mathbf{\Gamma}_{(1..m)(1..m)}^{-1} \mathbf{\Gamma}_{(1..m)0} - \mathbf{\Gamma}_{(m..n)0}^T \mathbf{\Gamma}_{(m..n)(m..n)}^{-1} \mathbf{\Gamma}_{(m..n)0}. \quad (5.32)$$

Combinando las ecuaciones 5.30 y 5.32 es fácil ver que

$$\mathbf{\Gamma}/\mathbf{\Gamma}_{(1..n)(1..n)} - \mathbf{\Gamma}_{(0..m)(0..m)}/\mathbf{\Gamma}_{(1..m)(1..m)} = -\mathbf{\Gamma}_{(m..n)0}^T \mathbf{\Gamma}_{(m..n)(m..n)}^{-1} \mathbf{\Gamma}_{(m..n)0}. \quad (5.33)$$

Recordando la ecuación 5.27, ahora puede ser reescrito como

$$\mathbf{\Gamma}/\mathbf{\Gamma}_{(m..n)(m..n)} = \mathbf{\Gamma}_{(0..m)(0..m)} + \begin{bmatrix} \mathbf{\Gamma}/\mathbf{\Gamma}_{(1..n)(1..n)} - \mathbf{\Gamma}_{(0..m)}/\mathbf{\Gamma}_{(1..m)} & 0 \\ 0 & 0 \end{bmatrix}. \quad (5.34)$$

Si la información marginal para el parámetro de referencia $\mathbf{\Gamma}/\mathbf{\Gamma}_{(1..n)(1..n)}$ estaba disponible, todos los términos de la ventana se pueden calcular con una complejidad que depende del tamaño de la ventana de optimización seleccionada.

En la práctica, lo que está disponible es el término, o más bien una aproximación de, $\mathbf{\Lambda}/\mathbf{\Lambda}_{(1..n)(1..n)} = \mathbf{\Sigma}_0^{-1}$. Usar este término en lugar de $\mathbf{\Gamma}/\mathbf{\Gamma}_{(1..n)(1..n)}$ induce alguna información adicional, específicamente cómo el parámetro de referencia se relaciona con el resto de los parámetros de la ventana por vías conectivas que pasan por encima de los parámetros omitidos. Esto hace que el primero sea un poco más restrictivo, pero dado que no se agrega información nueva o duplicada con respecto al problema original, el estimador aún debería ser conservador.

En la práctica, tal problema de optimización generalmente se construye a partir de un conjunto de funciones de error. El problema marginalizado se puede construir utilizando todas las restricciones de mediciones originales que relacionan al menos dos o más parámetros de la ventana. Además, se impone una restricción previa al parámetro de referencia de manera que produzca una función de error de la forma $\mathbf{x}_0 - \check{\mathbf{x}}_0$ con matriz de información asociada $\check{\mathbf{\Sigma}}_0^{-1} - \mathbf{\Gamma}_{(0..m)}/\mathbf{\Gamma}_{(1..m)}$. De este modo, la matriz de información $\mathbf{\Lambda}^\Pi$ para el problema de optimización marginal aproximado se convierte en

$$\mathbf{\Lambda}^\Pi = \left(\frac{\partial \begin{bmatrix} \mathbf{z}^T \\ \mathbf{x}_0^T \end{bmatrix}}{\partial \mathbf{x}} \right)^T \begin{bmatrix} \mathbf{\Sigma}_{z_W} & \mathbf{0} \\ \mathbf{0} & \check{\mathbf{\Sigma}}_0^{-1} - \mathbf{\Gamma}_{(0..m)}/\mathbf{\Gamma}_{(1..m)} \end{bmatrix}^{-1} \left(\frac{\partial \begin{bmatrix} \mathbf{z}^T \\ \mathbf{x}_0^T \end{bmatrix}}{\partial \mathbf{x}} \right), \quad (5.35)$$

y dado que la inversa de una matriz diagonal por bloques es una matriz diagonal con sus bloques invertidos, el producto se puede distribuir como

$$\mathbf{\Lambda}^{\Pi} = \mathbf{\Gamma}_{(0..m)(0..m)} + \frac{\partial \mathbf{x}_0^T}{\partial \mathbf{x}} \left(\mathbf{\Lambda}_{00} - \mathbf{\Lambda}_{0(1..n)}^T \mathbf{\Lambda}_{(1..n)(1..n)}^{-1} \mathbf{\Lambda}_{0(1..n)} \right) \frac{\partial \mathbf{x}_0}{\partial \mathbf{x}}. \quad (5.36)$$

que hasta la aproximación de $\mathbf{\Gamma}/\mathbf{\Gamma}_{(1..n)(1..n)}$ por $\check{\Sigma}_0^{-1}$ es la heurística derivada en la ecuación 5.34.

Se puede ver que el segundo factor solo afecta al bloque diagonal de $\mathbf{\Lambda}^{\Pi}$ correspondiente al parámetro \mathbf{x}_0 , mientras que el resto de la matriz de información para la ventana conserva los mismos elementos que los del bloque correspondiente $\mathbf{\Lambda}_{(0..m)}$ de la matriz de información original. Sin embargo, la información agregada por el prior contiene una gran cantidad de información sobre la matriz original.

Una segunda preocupación se debe al hecho de que la matriz de información resultante de la optimización realizada con un anterior único se usa para volver a estimar los parámetros marginales de las distribuciones de parámetros individuales. Cualquier pérdida de información causada por la aproximación de la verdadera marginal del problema de esta manera también se propagará a las nuevas estimaciones de parámetros. A su vez, estos pueden propagar aún más el error en optimizaciones futuras si se usan como priors, lo que puede hacer que las estimaciones de incertidumbre eventualmente divergan de las ideales a largo plazo.

Sin embargo, se puede decir que el estimador es conservador y parece funcionar bien en la práctica, como se mostrará en el Capítulo 7.

Por último es necesario mencionar brevemente la selección del parámetro de referencia de la ventana. Dado que sin más análisis cualquier parámetro de la ventana parece ser un candidato apropiado como referencia, una opción razonable sería aquella que intente mantener la incertidumbre estimada al mínimo. Dado que la información marginal aparece como un término positivo en la expresión final, el parámetro con la mayor cantidad de información se utilizará como referencia. O dicho de otro modo, el de menor incertidumbre.

5.3. Seguimiento de pose en tiempo constante

Un problema adicional que debe resolverse en complejidad constante está dado por la asociación de datos durante el proceso de seguimiento de pose. Es decir, dado un fotograma estéreo entrante y una predicción de pose incierta, establecer correspondencias de características locales detectadas en la imagen a marcas características del mapa. Por razones de rendimiento, es prohibitivamente costoso verificar las coincidencias contra todas las marcas de referencia existentes en el mapa, ya que la operación escala al menos linealmente con el tamaño (potencialmente ilimitado) del mapa. Por lo tanto, es necesario considerar solo una porción local delimitada del mapa que además debe poder recorrerse en tiempo constante.

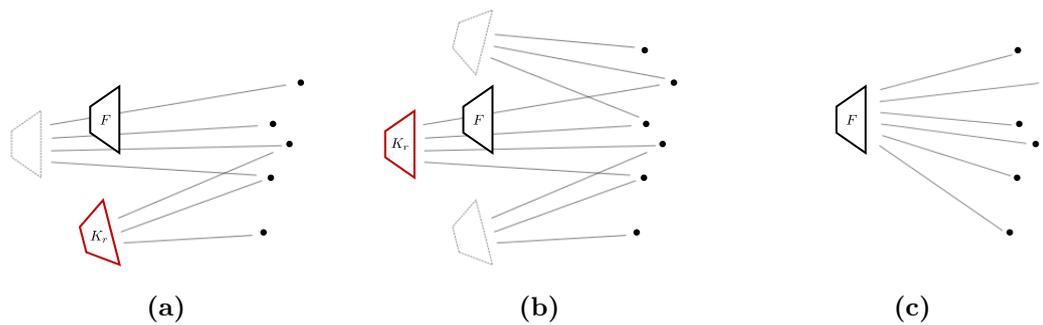


Figura 5.8: Proceso de actualización de la pose de referencia ante la llegada de un nuevo marco F . En la figura (a) puede observarse el marco de referencia anterior K_r , así como la ventana asociada al mismo. En (b) se elige, entre los elementos de dicha ventana, un nuevo marco de referencia por ser más próximo al marco F . La ventana local también es actualizada, y a partir de sus marcas del ambiente se deducen las asociaciones de mediciones desde el marco F , como puede apreciarse en la figura (c).

Esto se puede lograr aprovechando aún más la información de co-visibilidad para consultar las marcas de referencia visibles por los fotogramas clave más cercanos, ya que es probable que sean detectados en el cuadro actual. Para lograr esto, se mantiene un fotograma clave de referencia durante todo el proceso de SLAM que se utiliza como semilla para consultar marcas de referencia cercanas. Después de cada localización exitosa, este fotograma clave de referencia se actualiza seleccionando el fotograma clave más cercano al fotograma actual, eligiendo entre el último fotograma de referencia utilizado y todos los que se encuentran en su primer nivel de co-visibilidad.

El proceso de actualización del fotograma clave de referencia a la llegada de un nuevo fotograma se ilustra en detalle en la Figura 5.8. La figura 5.8a muestra el fotograma clave de referencia anterior K_r con sus vecinos co-visibles y los marcas de referencia asociadas, y un nuevo fotograma entrante F . Al llegar F , las distancias del mismo hacia los fotogramas clave co-visibles a K_r son revisadas, y tal como se muestra en la figura 5.8b se selecciona una nueva referencia K_r . Finalmente, las marcas de referencia asociadas al nuevo fotograma clave y su primer nivel de co-visibilidad son utilizadas para buscar asociaciones con las características locales de imagen del cuadro entrante F .

6 Sistema de navegación para hexápodos

Como indicado anteriormente, el objetivo principal de esta tesis es proporcionar un marco razonable sobre el cual se pueda ejecutar un algoritmo genérico de navegación en lazo cerrado, que es fundamental para que un robot móvil pueda realizar tareas autónomas con precisión. Los requisitos de servicios para tal algoritmo pueden ser arbitrariamente numerosos o complejos, pero un conjunto mínimo razonable puede definirse como consultar la ubicación del robot en el espacio y poder mover la plataforma dentro de él. Además, estos servicios deberían proporcionarse en forma abstracta, en este caso, una pose de 6 grados de libertad con una estimación de incertidumbre asociada y un comando de control de alto nivel en forma de velocidades angulares y lineales de 3 grados de libertad respectivamente. Una representación gráfica de este esquema de navegación básico se representa en la Figura 6.1.

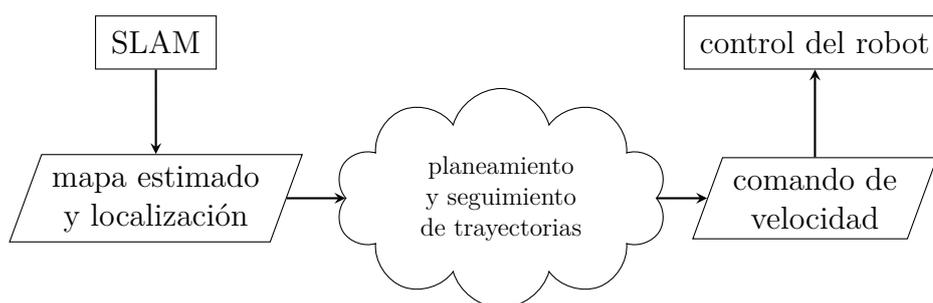


Figura 6.1: Diagrama de flujo que representa las relaciones entre un conjunto mínimo de servicios esenciales para el funcionamiento de un sistema de navegación de lazo cerrado.

Este capítulo propone una realización de los elementos del sistema presentados en la Figura 6.1, implementando así un sistema de navegación mínimo para un robot móvil hexápodo. Las instancias de estos módulos se eligieron específicamente para realizar navegación autónoma en tiempo real de robots hexápodos operando en terrenos difíciles. La instancia del esquema correspondiente se muestra en la Figura 6.2 donde cada capa fue seleccionada cuidadosamente y diseñada para cumplir con los requisitos necesarios.

Como sistema de localización se seleccionó el sistema de SLAM visual S-PTAM [3] debido a su desempeño de vanguardia y la familiaridad del autor con la base de

código subyacente. Se amplió mediante la implementación de los algoritmos analizados en los capítulos 4 y 5. También se ha incluido un modelo predictivo basado en mediciones inerciales, para hacer frente a los movimientos repentinos inherentes de una plataforma de hexápodos que navega en terrenos irregulares.

Como módulo de control, se seleccionó el Adaptive Motion Gait (AMG) por su simplicidad en el manejo de terrenos irregulares. Aprovechando la retroalimentación de error de posición de los servos disponibles en la plataforma hexápoda utilizada, se puede estimar el contacto con el suelo sin la necesidad de más sensores. Como resultado, el cuerpo se mantiene bastante estable durante la navegación.

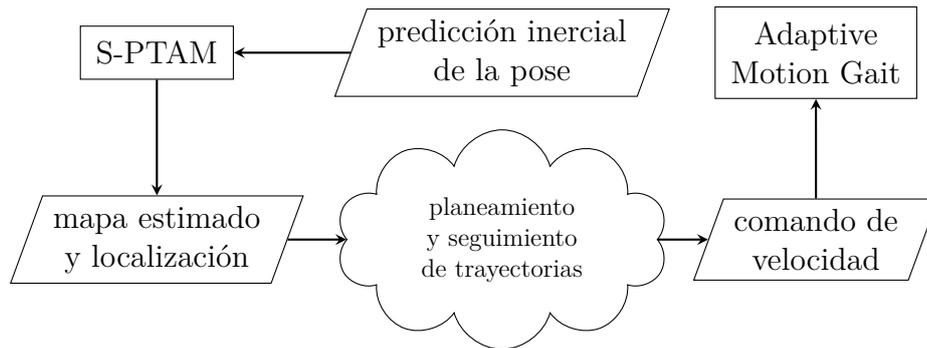


Figura 6.2: Diagrama de flujo que representa una instancia de un conjunto mínimo de servicios en el contexto de un sistema de navegación de lazo cerrado.

A continuación, se presenta una explicación detallada de los sistemas S-PTAM y AMG, y sus detalles de implementación relevantes que permiten que el sistema se ejecute en hardware limitado en tiempo real.

6.1. S-PTAM

S-PTAM [85, 3] es un sistema de SLAM visual basado en optimización que se basa en características locales de imagen para la localización de la cámara y el mantenimiento de una reconstrucción global del entorno. Sigue el enfoque adoptado por Parallel Tracking and Mapping (PTAM) [33, 34], donde el procedimiento de optimización de mapas más lento se ejecuta en un hilo diferente al de la estimación de pose de la cámara en tiempo real, aprovechando así las ventajas de las arquitecturas de CPU multi-core. Además, un módulo de detección y corrección de ciclos se ejecuta en un proceso separado para evitar así la duplicación de mapa y garantizar la consistencia global. Las marcas de referencia y poses asociadas son los únicos recursos compartidos entre los hilos, y se debe cuidar el manejo adecuado de la concurrencia, especialmente en los elementos ubicados en las inmediaciones del robot. La figura 6.3 muestra el esquema general de los componentes principales de S-PTAM.

Al inicio, el sistema comienza con la cámara en una pose arbitraria seleccionada arbitrariamente respecto de un marco de referencia global (por convención, el origen del

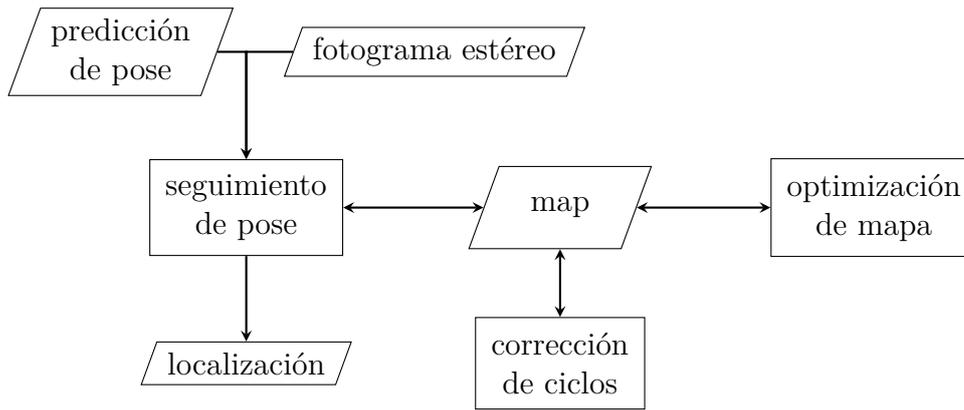


Figura 6.3: Diagrama de flujo de alto nivel del sistema S-PTAM. Cada proceso representado se ejecuta en su propio hilo. El único recurso compartido entre ellos es el mapa.

espacio euclidiano) y un mapa vacío, que se inicializa inmediatamente triangulando las características apareadas en el primer par de imágenes estéreo entrantes. Tenga en cuenta que al usar cámaras estéreo se evita el problema típico de *bootstrapping* monocular. De aquí en adelante, para cada nuevo par de imágenes estéreo entrantes, se requiere calcular una nueva estimación de pose. Dado que se espera que un sistema de localización responda en tiempo real, la estimación debe calcularse dentro de una complejidad computacional limitada razonable. Inmediatamente se extraen las características locales de imagen y se aparean con las marcas de referencia existentes en el mapa. Estas correspondencias se pueden usar para estimar la posición de la cámara con respecto al mapa minimizando los errores de re-proyección de la imagen [74], como se explica en la Sección 4.3.2.

Dado que aumentar el mapa con nuevos elementos es en gran medida redundante para fotogramas espacialmente próximos, e implica mayores costos computacionales durante futuras optimizaciones, no se realiza para cada fotograma entrante. Más bien, al cumplirse una política de criterios de novedad, el fotograma estéreo puede ser seleccionado como un fotograma clave. Cuando esto sucede, las características locales de la imagen que no tienen correspondencias con marcas del ambiente se triangulan a partir del par estéreo, y el fotograma clave, así como las marcas del ambiente recién trianguladas, se agregan al mapa. En la implementación actual, un fotograma se promueve a un fotograma clave si se excede un cierto umbral de distancia con respecto al fotograma clave de referencia, ya sea en distancia lineal o diferencia angular. También se probaron otras políticas, como el área de cobertura de características correspondidas respecto al área de la imagen, pero resultaron demasiado estrictas.

En un hilo paralelo, el procedimiento de optimización del mapa refina periódicamente un vecindario local del mapa mientras se incorporan datos nuevos, para mantenerlo lo más consistente posible. Al mismo tiempo, el proceso de corrección de ciclos está

continuamente verificando posibles correspondencias, que al ser detectadas activan otro procedimiento de optimización, aunque de tamaño potencialmente ilimitado, en un intento de enmendar los errores de deriva para mantener el mapa consistente.

Para lograr un rendimiento en tiempo real en plataformas de hardware con bajos recursos, es crucial identificar las operaciones del sistema de Visual SLAM que son computacionalmente exigentes. La Figura 6.4 muestra una vista más detallada de los módulos principales tal como se presenta en la Figura 6.3, junto con sus procedimientos clave. A continuación, se discuten algunos detalles de implementación para la estructura de datos central del sistema S-PTAM, el mapa, así como para cada uno de estos procesos y sus principales operaciones.

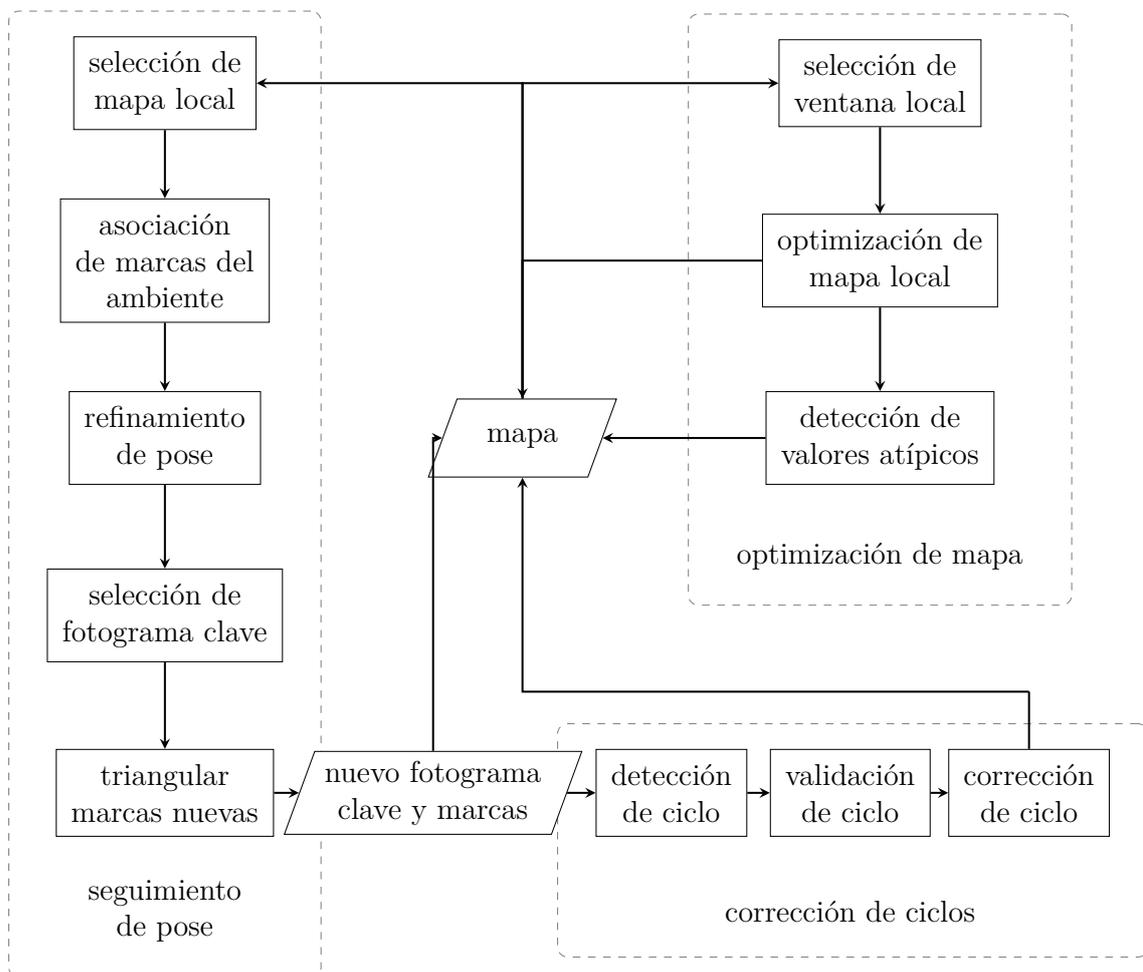


Figura 6.4: Diagrama de flujo que describe las principales tareas del sistema S-PTAM y sus relaciones. Los hilos independientes están agrupados por cajas amarillas.

6.1.1. El mapa

El mapa de co-visibilidad es la estructura de datos central alrededor de la cual se construye el sistema S-PTAM. En su núcleo, presenta una típica estructura de grafo bipartito en la que los dos tipos de nodos representan poses estimadas de fotogramas clave y marcas de referencia del ambiente detectadas. Estos a su vez están relacionados por aristas que representan las mediciones de marcas características en las imágenes de la cámara del fotogramas clave, y las estimaciones de movimiento entre poses de fotogramas sucesivos. Esta estructura básica del mapa es aumentada agregando otra colección de aristas que representa los grados de co-visibilidad entre las poses de fotogramas clave cercanos. El peso de una de estas aristas entre dos poses, indica el número de marcas de referencia en común que se detectaron desde ambas. Estas aristas no agregan ninguna información nueva al grafo, pero la resumen para que sea accesible en tiempo constante.

El mapa se implementa como una colección dinámica de nodos vinculados, donde cada nodo mantiene enlaces a los elementos directamente relacionados. Este enfoque descentralizado de representación gráfica permite estrategias de bloqueo eficientes cuando diferentes procesos trabajan sobre diferentes secciones del mapa. Un ejemplo de tal situación se representa en la Figura 6.5. Allí, el módulo de seguimiento está realizando la estimación actual de la pose de la cámara, mientras que el módulo de optimización local está realizando correcciones en el vecindario. Al mismo tiempo, el módulo de corrección de ciclos está corrigiendo los fotogramas clave más antiguos, que no están siendo utilizados activamente por ningún otro módulo, para garantizar la coherencia del mapa global.

Sin embargo, el acceso simultáneo a elementos de mapa individuales aún podría ser requerido por varios procesos al mismo tiempo. Ocasionalmente, cuando el módulo de seguimiento de pose declara un fotograma como fotograma clave, se insertará una nueva pose en el mapa, lo que requiere actualizar las relaciones en las marcas de referencia asociadas y las poses co-visibles, que a su vez pueden estar en uso por otras operaciones concurrentes. Además, el hilo de optimización del mapa actualiza periódicamente los valores de los parámetros cercanos y elimina los valores atípicos, muchos de los cuales se utilizan de forma continua para rastrear los marcos entrantes. Finalmente, el proceso de corrección de ciclos puede estar realizando una optimización de ciclo detectada en una zona cercana a la actual, por lo que también intentará actualizar los valores de los parámetros locales. Queda claro que un sistema de este tipo también implica un acceso altamente concurrente a elementos específicos del mapa, particularmente aquellos que están cerca de la posición actual del robot.

Un bloqueo de escritura individual para un elemento se otorga solo si todos los elementos relacionados, es decir, todos aquellos que tienen referencias a él, también pueden ser bloqueados. Esto evita el bloqueo de la estructura completa para el acceso a cada elemento específico, y además garantiza que cuando la modificación del elemento implique actualizaciones en sus relaciones, éstas también se pueden

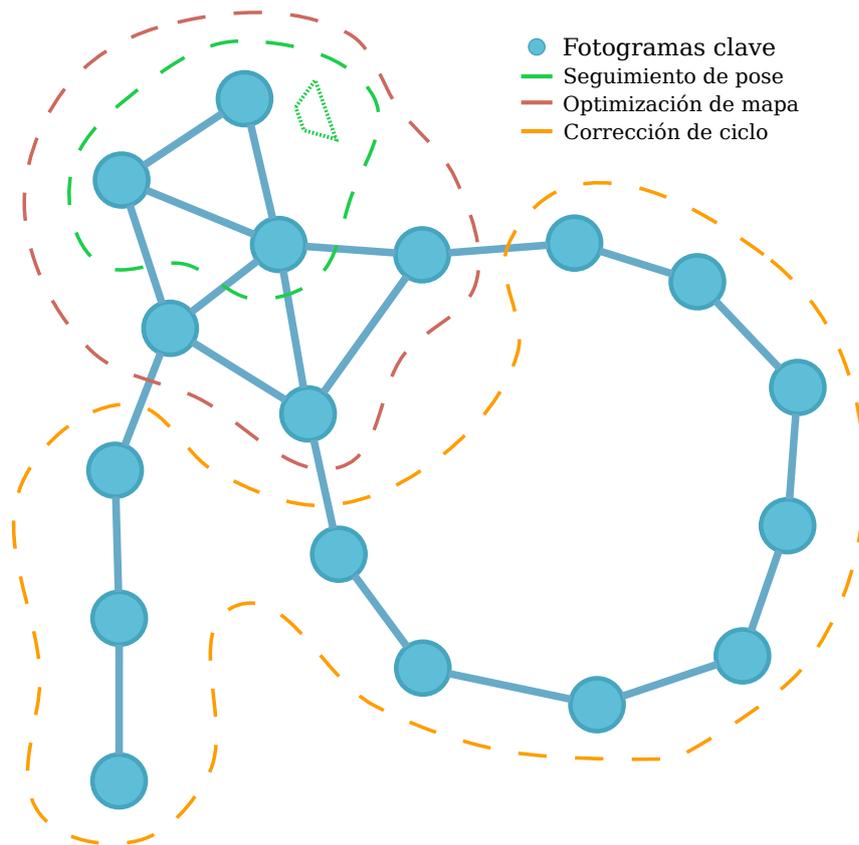


Figura 6.5: Ejemplo de segmentación de mapas en una situación en la que las subprocesos concurrentes utilizan diferentes áreas de forma activa.

realizar de manera segura, manteniendo así la consistencia de la estructura en todo momento. Para evitar abrazos mortales, primero debe adquirirse un bloqueo superior a nivel del mapa para luego bloquear cualquiera de los elementos específicos, el cuál es liberado inmediatamente después de que se adquiere este último.

Además, la estructura permite un recorrido de eficiente de la vecindad en términos de visibilidad, lo cual es crítico para el rendimiento de los algoritmos explicados en el Capítulo 5. Allí se asume que visitar todos los elementos directamente relacionados no requiere una complejidad mayor que la cantidad de dichos elementos. Este es efectivamente el caso, ya que cada nodo mantiene una lista actualizada de referencias a todos los elementos directamente relacionados. Aunque cada vez que se insertan o eliminan mediciones de marcas de referencia, las relaciones de co-visibilidad deben actualizarse, la complejidad se considera limitada por una constante, ya que ya se asumió que el conjunto de fotogramas clave adyacentes se mantiene esparso debido a la política de selección de fotogramas clave basada en distancia.

6.1.2. Seguimiento de pose

El módulo de seguimiento de pose está a cargo de la tarea más importante en el sistema SLAM, en torno a la cual se construye todo lo demás: proporcionar continuamente, y en tiempo real, estimaciones precisas de la pose del robot con respecto al entorno mapeado. Lo hace minimizando los errores de re-proyección entre las características locales de la imagen y sus correspondientes marcas de referencia del ambiente asociadas.

Como las optimizaciones numéricas tienden a caer en mínimos locales, generalmente se requiere una buena estimación inicial para alcanzar el óptimo global. Normalmente, la última posición conocida de la cámara se aumenta agregando un movimiento estimado, que puede ser proporcionado por sensores propioceptivos adicionales (p.e. IMU, codificadores de ruedas, etc.) o, si no hay otros sensores de movimiento disponibles, por un modelo de movimiento artificial (p.e. un decaimiento constante de la velocidad).

Esta estimación inicial también se usa para mantener una referencia rápida a los elementos del mapa que están espacialmente cerca del robot. Bajo el supuesto de que el robot se mueve en forma continua (es decir, no es posible “teletransportarse”) y que las muestras de la pose se toman lo suficientemente rápido, se puede obtener el fotograma clave más cercano a la posición actual estimada consultando la vecindad del fotograma clave más cercano a la última pose estimada. La selección del vecindario local del mapa implica un recorrido BFS sobre el grafo de co-visibilidad que, a su vez, se puede limitar para seleccionar hasta un número constante de elementos.

A continuación, deben establecerse las correspondencias entre las características locales de la imagen detectadas y las marcas de referencia del mapa en la vecindad local. Esto se logra proyectando las posiciones de dichas marcas de referencia sobre los planos de imagen en la pose estimada y realizando una búsqueda de vecino más cercano en el espacio del descriptor de imagen. Esta búsqueda se ve facilitada por el hecho de que las características locales de la imagen se combinan en el momento de la detección en una estructura de grilla bidimensional, lo que permite consultar las características que se encuentran en una zona de tamaño arbitrario de manera muy eficiente (Figura 6.6).

Las correspondencias estimadas se pueden considerar como un conjunto de restricciones sobre el parámetro de pose, que se pueden refinar mediante la optimización numérica. Esto se hace usando un algoritmo de Levenberg-Mardquardt para minimizar el error cuadrático medio de proyección de las posiciones de marcas de referencia del mapa con respecto a sus características locales de la imagen correspondientes. También se estableció un número máximo de iteraciones para garantizar un límite superior en el tiempo de cómputo requerido. En la práctica, un límite superior de diez iteraciones fue suficiente para abarcar cualquier optimización exitosa. Se dice que una optimización converge con éxito si entre dos iteraciones consecutivas la diferencia en los errores calculados cae por debajo de un pequeño umbral de tolerancia predeterminado.

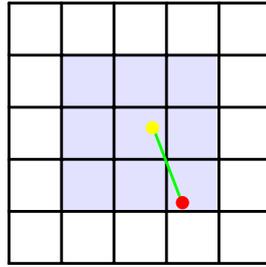


Figura 6.6: Ejemplo de asociación de datos asistidos por la grilla. Una proyección contra el plano de imagen de la cámara estimada (punto amarillo) se compara con características locales de imagen dentro de un vecindario de tamaño 3x3 (azul claro). Si se encuentra una buena correspondencia (punto rojo) se establece una relación.

Una vez que la pose actual se ha estimado correctamente, se debe tomar una decisión sobre si el marco actual debe seleccionarse como un fotograma clave para aumentar el entorno mapeado o no. Una heurística es empleada para verificar si hay suficientes datos nuevos que agregar y / o si el proceso de seguimiento está a punto de perder la pista del mapa, es decir, la posición de referencia está demasiado lejos. Específicamente, si se excede un cierto umbral de distancia respecto del el fotograma clave de referencia, ya sea en distancia lineal o diferencia angular, se crea un nuevo fotograma clave. Las características locales de la imagen que no tienen correspondencias con marcas del ambiente se triangulan a partir del par estéreo, y el fotograma clave, así como las marcas recién generadas, se agregan al mapa.

6.1.3. Optimización del mapa

El módulo de optimización de mapa se encarga de mantener la consistencia global y la precisión del mapa cada vez que se incorporan nuevos datos. Dado que el entorno mapeado es potencialmente ilimitado, realizar una optimización completa de todo el mapa se vuelve prohibitivamente costoso. Como es poco probable que los elementos que tienen poca o ninguna relación con el área actualmente activa se vean afectados por los cambios, se puede definir una ventana de optimización de tamaño constante como se explicó anteriormente en el Capítulo 5. Esto permite que el módulo realice optimizaciones periódicas pero pequeñas que mantienen el área local precisa, y por lo tanto mejoran localizaciones futuras realizadas por el módulo de seguimiento.

La optimización de cuadrados mínimos se realiza utilizando un algoritmo de Levenberg-Marquardt, en particular utilizando la biblioteca *ceres* [86] disponible libremente. El número de iteraciones está limitado por una constante arbitraria, que además del número fijo de variables y restricciones mantiene el tiempo de optimización acotado. De manera análoga al módulo de seguimiento de pose, un límite superior de diez iteraciones fue suficiente para realizar una optimización exitosa.

Posteriormente, se realiza un esfuerzo para mejorar la calidad del mapa mediante la detección y descartar mediciones que no se conciben con la solución encontrada. Como se explica en la Sección 4.5, se consideran valores atípicos de mediciones aquellas proyecciones que caen fuera de un intervalo de confianza arbitrario (Sección 3.2.3) de las densidades normales independientes que rigen la distribución de errores en cada dimensión (los ejes de la imagen) de la característica local de imagen. Se utiliza un intervalo de confianza típico del 95 % para los experimentos descritos en esta tesis. Se debe tener cuidado para elegir un umbral apropiado para la distribución chi-cuadrado de cada medición, ya que las mediciones estereo y monocular tienen una dimensionalidad diferente.

6.1.4. Corrección de ciclos

Manejar entornos grandes requiere reconocer lugares ya visitados y optimizar el mapa y la trayectoria para reducir la deriva acumulada y mantener la consistencia global. Con este fin, el módulo de cierre de ciclos revisa cada nuevo fotograma clave agregado al mapa para detectar posibles ciclos. Esto se hace de manera eficiente mediante la descripción de cada cuadro de imagen a partir de una bolsa de palabras binarias [87] (BoW) utilizando un vocabulario visual entrenado previamente de manera independiente. Así se va generando incrementalmente una base de datos de lugares visitados, al insertar un descriptor BoW para cada fotograma clave agregado en una estructura de datos orientada a la búsqueda llamada *árbol de vocabulario* [88]. Luego, para cada nuevo fotograma entrante, esta base de datos puede consultarse de manera eficiente para detectar lugares anteriores que tienen palabras similares al actual.

Si se encuentra un fotograma clave K_l que representa un candidato de bucle adecuado para un fotograma clave entrante K_c , se activa el algoritmo de corrección de bucle. La Figura 6.7 ilustra las diferentes etapas involucradas en el proceso. Inicialmente, una transformación relativa l_cT entre ambos extremos del ciclo es estimada mediante un método P3P utilizando ambos pares de imágenes estereo. Luego las mediciones de marcas de referencia son marginalizadas para obtener un grafo de poses del ciclo y así acelerar el proceso de optimización (figura 6.7b). La corrección de la primera etapa se realiza al propagar l_cT hacia atrás en el grafo de pose mediante una simple interpolación (figura 6.7c). Finalmente, se realiza una optimización de cuadrados mínimos al grafo de poses, y luego la posición de cada marca de referencia se actualiza utilizando la misma corrección que se aplicó al fotograma clave que originalmente lo trianguló durante el proceso de corrección del ciclo (figura 6.7d).

Para terminar, la estructura del mapa se actualiza con la información corregida en dos etapas. Primero, los parámetros no involucrados en la optimización del mapa local y las operaciones de seguimiento de pose se descargan al mapa. Luego, una barrera de sincronización es bloqueada y ambas operaciones concurrentes se detienen momentáneamente mientras el área activa del mapa se actualiza con la corrección del ciclo.

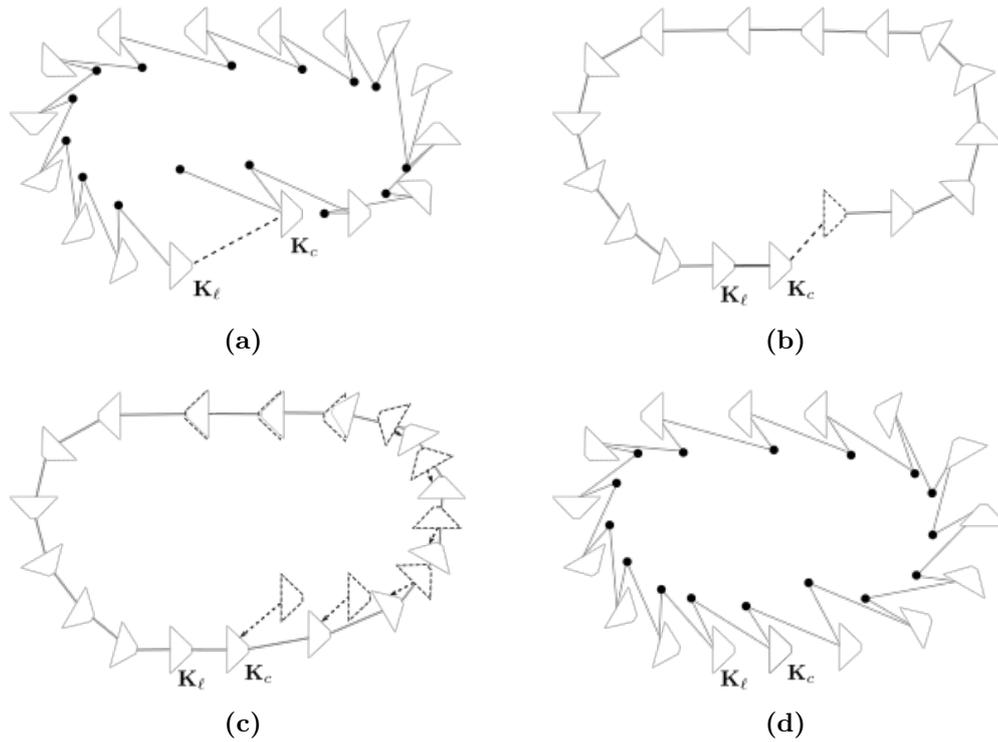


Figura 6.7: Etapas del algoritmo de corrección de ciclos. K_c representa el fotograma clave actual y K_l el candidato de ciclo.

6.2. Adaptive Motion Gait

Los robots con patas, como los robots de hexápodos, exhiben mayores capacidades de navegación en terrenos accidentados y desestructurados que sus contrapartes con ruedas. Sin embargo, estas capacidades de movimiento vienen acompañadas de una mayor complejidad de control, que se debe principalmente al alto número de grados de libertad (DoF). El robot hexápodo utilizado en esta tesis tiene un total de 18 DoF, tres actuadores por cada una de las seis piernas. Por lo tanto, la búsqueda de soluciones óptimas en el espacio de configuración de las piernas se vuelve computacionalmente prohibitiva. Una solución más eficiente es generar un patrón de caminata o *gait* [89].

Los patrones de caminata simples de onda periódica, donde se alternan periódicamente pares de patas, pueden ser muy eficiente en terrenos planos donde todas las patas que soportan al robot se encuentran en el mismo plano. Además, sobre una superficie plana, es suficiente elevar cada pata a la altura mínima y moverla hacia adelante. Sin embargo, en terrenos irregulares y difíciles, el robot debe atravesar pequeños obstáculos y los efectores finales de las piernas pueden terminar a diferentes alturas. Las piernas pueden incluso perder el apoyo del suelo o atascarse, evitando que el robot se mueva en la dirección solicitada. Por lo tanto, los obstáculos simples

pueden ser difícilmente transitables usando un patrón de caminata simple en forma de lazo abierto. Esto se puede superar cerrando el circuito de control al considerar información sensorial adicional cuando se genera y ejecuta el patrón de movimiento.

Mrva y Faigl [24] proponen que la información de posición brindada por muchos servomotores también puede usarse para asegurar que las patas individuales están dando soporte efectivo al robot. Esta es una configuración bastante minimalista que no requiere de sensores adicionales como en otros trabajos [90, 91]. En ella, la detección táctil se basa en medir la fuerza actuando en contra del servomotor que resulta del contacto con el pie cuando se encuentra con un obstáculo. La fuente de la fuerza original es el torque del servomotor, pero se considera que este está relacionado linealmente con el error de posición del servo, como se muestra en la Figura 6.8. Por lo tanto, la fuerza de actuación en el servo puede estimarse analizando el error de posición con respecto a la posición objetivo actual. Además, se puede definir un valor de umbral $e_{threshold}$ que, cuando se excede, indica una carga alta en el servo, y por lo tanto, una fuerza de actuando en contra, que en este caso señala una colisión con un obstáculo. Es importante tener en cuenta que la dinámica de los robots no es tenida en cuenta para establecer el umbral de contacto, pero el algoritmo parece funcionar bien en la práctica una vez que se establece un valor de umbral fijo para una plataforma y velocidad predefinidas.

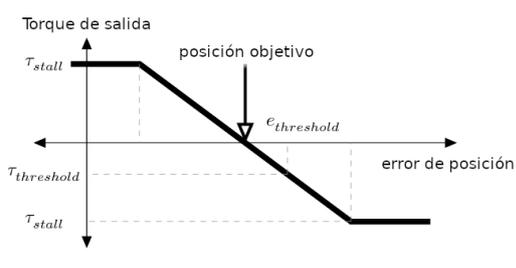


Figura 6.8: Estimación de la relación entre el torque de salida del servo y el error de posición mientras se controla la posición del objetivo.

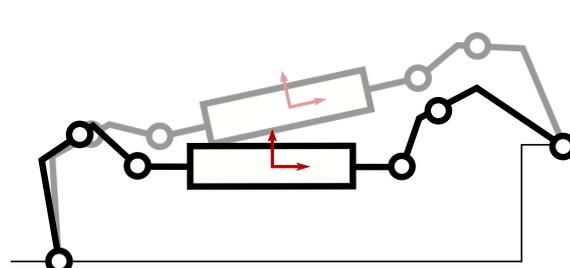


Figura 6.9: Visualización de la nivelación del cuerpo de un robot hexápodo durante un patrón de caminata adaptativo (adelante) y tradicional (en marca de agua).

El algoritmo de *Adaptive Motion Gait* [24] (AMG) es una adaptación de los patrones clásicos que aprovecha la detección de contacto con el suelo para establecer nuevos puntos de apoyo durante la planificación del patrón de caminata. El objetivo es mantener el peso del robot distribuido uniformemente entre todas las piernas, es decir, todas deben soportar la misma (o similar) carga para sostener al robot en un estado estable. La Figura 6.9 muestra una comparación entre el robot hexápodo pisando un obstáculo alto durante un patrón de caminata adaptativo (adelante) y tradicional (en marca de agua).

6.3. Algoritmo de navegación

Se definió un algoritmo de navegación autónoma simple sobre el sistema propuesto como una prueba de concepto para incentivar el desarrollo de métodos más complejos en un futuro. Usando la localización provista por el sistema S-PTAM, el robot es comandado a seguir una trayectoria geométrica simple siguiendo una secuencia de puntos de ruta descritos en un espacio bidimensional paralelo al suelo. El controlador de navegación envía comandos de velocidad lineal y angular al robot basándose en un control proporcional de la posición que mide la distancia al siguiente objetivo utilizando la localización actual.

Dado que el terreno está poblado por obstáculos que simulan un terreno accidentado, la estimación provista por el sistema S-PTAM deberá ser lo suficientemente buena como para proporcionar una trayectoria 2D con respecto al plano del suelo, a pesar de atravesar también terreno vertical. El sistema deberá responder lo suficientemente rápido para ajustar las velocidades en consecuencia y llegar a los puntos de destino de forma precisa y con la velocidad adecuada. Por otro lado, se requerirá que el sistema AMG sea capaz de navegar un conjunto de obstáculos y terrenos irregulares, y mantener mientras tanto la plataforma robótica lo mejor estabilizada posible para minimizar movimientos erráticos que puedan deteriorar las estimaciones computadas por el sistema de localización.

7 Experimentos

7.1. Entorno simulado

Un marco de simulación y prueba fue desarrollado específicamente para evaluar aspectos particulares del algoritmo de Bundle Adjustment constante bajo condiciones precisamente controladas. La configuración del entorno consiste en especificar una trayectoria (por ejemplo, velocidad lineal y angular constante) para un robot simulado, así como parámetros que controlan el número de marcas de referencia (rastreadas y creadas) que deben ser visibles en cada cuadro. Las distribuciones de ruido blanco gaussiano se pueden configurar para las estimaciones de pose iniciales en cada paso, así como para las características locales de imagen detectadas.

El algoritmo de generación se describe en Algoritmo 7.1. Los parámetros son el número de cuadros estéreo para generar (`n_frames`), el número deseado de marcas de referencia observadas en cada cuadro (`max_landmarks`), la velocidad constante del vehículo (`velocity`), la velocidad de cuadros (`dt`) y los parámetros de ruido para la simulación de características. Estos fueron elegidos como ruido gaussiano blanco con $\sigma = 1\text{px}$.

Algorithm 7.1 Generate a simulated trajectory.

```
1: pose  $\leftarrow$  0
2: frame_id  $\leftarrow$  0
3: landmarks  $\leftarrow$   $\emptyset$ 
4: while frame_id < n_frames do
5:   matched_landmarks, matched_features  $\leftarrow$  match(pose, landmarks)
6:   n_new  $\leftarrow$  max(0, max_landmarks - #matched_landmarks)
7:   new_landmarks, new_features  $\leftarrow$  triangulate(pose, n_new)
8:   save(frame_id, new_features, matched_features)
9:   landmarks  $\leftarrow$  landmarks  $\cup$  new_landmarks
10:  pose  $\leftarrow$  pose + velocity  $\times$  dt
11:  frame_id++
12: end while
```

La rutina `match` selecciona el subconjunto de marcas de referencia existentes que caen en el campo visual de la cámara en la posición dada. Estas marcas de referencia luego se proyectan sobre los planos de la imagen y se contaminan con ruido gaussiano blanco para simular el conjunto de características locales de imagen observadas. Si

el número de características detectadas es menor que el número deseado de marcas de referencia observadas por fotograma (`max_landmarks`), las marcas restantes se crean aleatoriamente dentro del frustum de la cámara. Estos también se proyectan en los planos de la imagen y se contaminan con ruido gaussiano blanco para simular las características utilizadas para generar nuevas marcas de referencia. Para cada cuadro, el algoritmo genera un conjunto de características que coinciden con marcas de referencia ya existentes y otro conjunto de características que deben utilizarse para triangular nuevas. Es importante darse cuenta de que la asociación entre las características y las marcas de referencia detectadas está dada y es perfecta, lo cuál no es el caso en un escenario real.

Por otro lado, la rutina de simulación recibe las características de imagen simuladas para cada fotograma estéreo entrante y una secuencia de mediciones de odometría que se deducen de las poses conocidas. En cada cuadro, los datos de odometría son contaminados por ruido gaussiano blanco para simular errores inducidos por el proceso de predicción de movimiento. Los valores específicos para estas pruebas fueron $\Sigma_v = \mathbf{I}_3\sigma_v$ con $\sigma_v = 0,01$ para velocidad lineal y $\Sigma_\omega = \mathbf{I}_3\sigma_\omega$ con $\sigma_\omega = 0,1$ para la velocidad angular. En ambos casos, los ruidos son independiente por eje y no están correlacionados. Esta predicción de pose se usa como una solución inicial para la pose de la cámara en el próximo problema de optimización. Las asociaciones entre las marcas de referencia y las nuevas características locales de imagen son proporcionadas por el marco de simulación. Se computa una ventana de optimización, que incluye el nuevo fotograma y las restricciones a marcas del ambiente, y el Bundle Adjustment se lleva a cabo proporcionando un conjunto actualizado de parámetros.

El objetivo principal de ejecutar SLAM en estos entornos es comparar el rendimiento de la heurística de estimación de la distribución marginal propuesta en Capítulo 5 con la solución de canónica en condiciones ideales. Para este propósito se simulan dos escenarios simples pero representativos: una trayectoria lineal sin cierres de lazo y una circular que cierra un lazo.

El primer escenario simula una cámara estéreo que se mueve en línea recta con velocidad constante durante un período prolongado. Se eligió el experimento para exhibir la evolución de las covarianzas en un entorno puramente odométrico, donde no se producen cierres de ciclo. Se espera que las covarianzas crezcan a cada paso, y que las covarianzas computadas por la heurística de prior implementada crezcan de manera similar a las computadas por la solución de Bundle Adjustment canónica. Además, el primero no debería ser más informativo con respecto al último. Los resultados de las estimaciones de covarianza para este experimento se pueden ver en la Figura 7.1 donde además se observa el comportamiento esperado.

El segundo escenario simula un robot que realiza una trayectoria circular y, finalmente, vuelve a visitar el origen y los primeros fotogramas de la trayectoria. Esta configuración permite estudiar el comportamiento de las estimaciones de densidad cuando se cierran ciclos respecto de secciones mucho más antiguas del mapa, donde datos muchos más informativos se relacionan de repente con los fotogramas de la re-

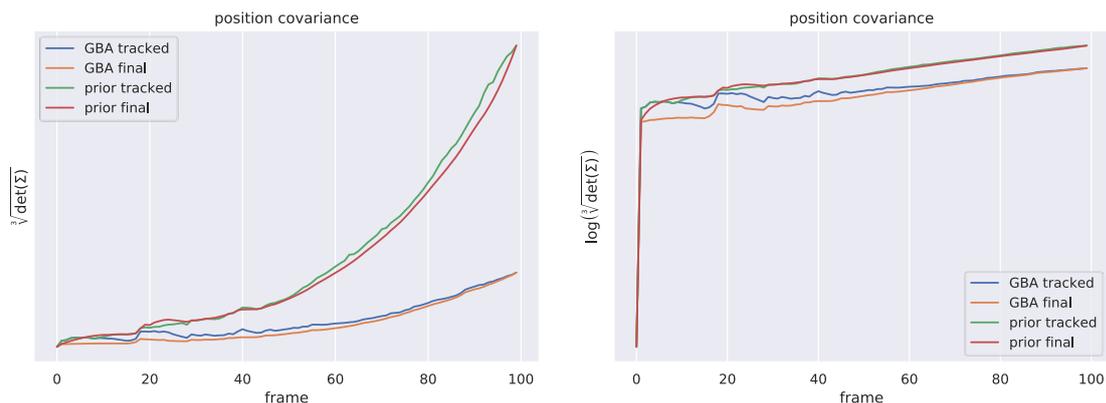


Figura 7.1: Volúmenes comparativos de covarianza de posición para una trayectoria lineal simulada de 100 cuadros. Se muestran dos métodos de selección de ventana: Optimización global (GBA) y una ventana de visibilidad de tamaño 10 con prior (prior). Para cada uno de estos, se muestran las poses estimadas (tracked), así como las poses optimizadas finales (final).

gión actualmente activa. El experimento se diseñó específicamente para justificar la elección del parámetro de referencia, que se supone maximiza el valor informativo de la ventana. La Figura 7.2 muestra las covarianzas resultantes para los experimentos que utilizan la ventana propuesta, así como otros que realizan Bundle Adjustment completo en cada paso. Cuando una región más informativa del mapa se redescubre y se relaciona con la ventana activa, la covarianza de la pose estimada cae inmediatamente cuando se realiza una GBA. Para la ventana propuesta, la covarianza también se reduce aunque unos pocos fotogramas más tarde, cuando la región revisitada se vuelve lo suficientemente visible como para que el fotograma actual la considere entre los 10 fotogramas clave de la ventana de optimización.

Las figuras 7.3 y 7.4 representan los momentos antes y después de que una parte más antigua del mapa sea redescubierta por primera vez por ambos métodos. Respectivamente, el método que realiza el Bundle Adjustment completo en cada paso y el que utiliza la ventana y estrategia de marginalización propuestas. En este punto, se establecen las correspondencias entre el fotograma actual y los primeros de la secuencia, vinculando efectivamente los fotogramas más nuevos a datos mucho más informativos, dejando caer sus propias covarianzas marginales a un valor mucho menor que el que venían arrastrando. El efecto también se puede observar en las figuras Figura 7.1 y Figura 7.2, donde la covarianza para la pose actual cae significativamente al procesar el fotograma decisivo. El método que realiza GBA es claramente superior, ya que cierra el ciclo más rápido y actualiza inmediatamente las covarianzas marginales de todo el mapa. Sin embargo, ya se comentó que este enfoque no escala bien en complejidad cuando es utilizado en aplicaciones de tiempo real. También se puede observar cómo las covarianzas más grandes para las estimaciones de la pose final son las que se encuentran más alejadas de las más informativas (el origen), y

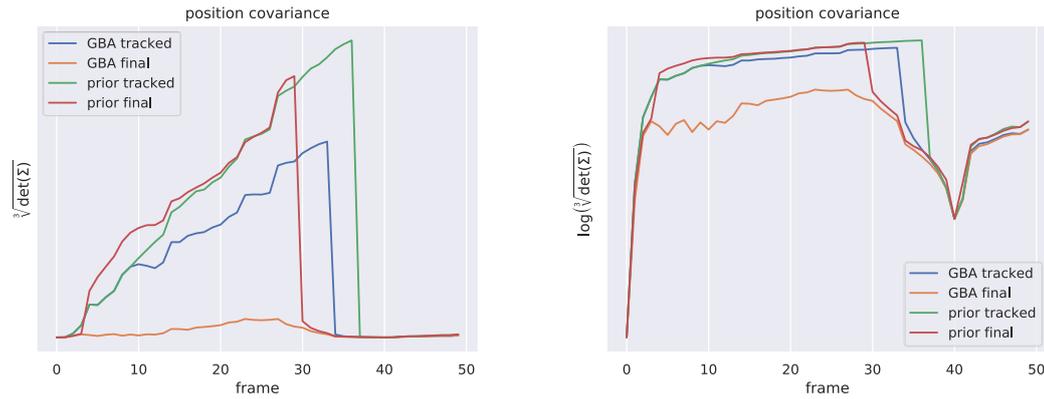


Figura 7.2: Volúmenes comparativos de la covarianza de posición para una trayectoria circular simulada de 100 cuadros. Se muestran dos métodos de selección de ventana: Optimización global (GBA) y una ventana de visibilidad de tamaño 10 con prior unario. Para cada uno de estos, se muestran las poses estimadas en cada paso, así como su estado de la trayectoria al terminar la secuencia.

la información se propaga a través del grafo de visibilidad independientemente de la dimensión temporal.

7.2. Plataforma robótica

Aunque los resultados de los experimentos en entornos simulados que se muestran en la sección anterior sugieren un rendimiento razonable, el despliegue y la evaluación en plataformas robóticas reales son el objetivo final cuando se apunta a aplicaciones en el mundo real. En particular, este trabajo se centró en el desarrollo de un sistema de localización para robots hexápodos, que pueden sufrir de movimientos repentinos y arbitrarios de la cámara debido a irregularidades en el suelo y deslizamientos. Por lo tanto, se desarrolló una plataforma robótica capaz de atravesar estos tipos de terrenos y ejecutar un sistema Visual SLAM integrado en tiempo real.

La plataforma elegida es el robot hexápodo PhantomX Mark II desarrollado por Trossen Robotics presentado en la Figura 7.5. La plataforma proporciona un marco con seis piernas de tres grados de libertad cada una. Cada articulación está formada por un servo DYNAMIXEL AX-12A que proporciona la retroalimentación de posición necesaria para alimentar el algoritmo Adaptive Motion Gait (Sección 6.2). Además, la plataforma ofrece espacio suficiente para montar equipos adicionales en la parte superior, mientras que una batería de 12V cabe cómodamente dentro del marco del cuerpo.

Como unidad de procesamiento a bordo se eligió una computadora ODROID-XU4, que brinda un equilibrio razonable entre velocidad de procesamiento, tamaño y consumo de energía. Las características incluyen

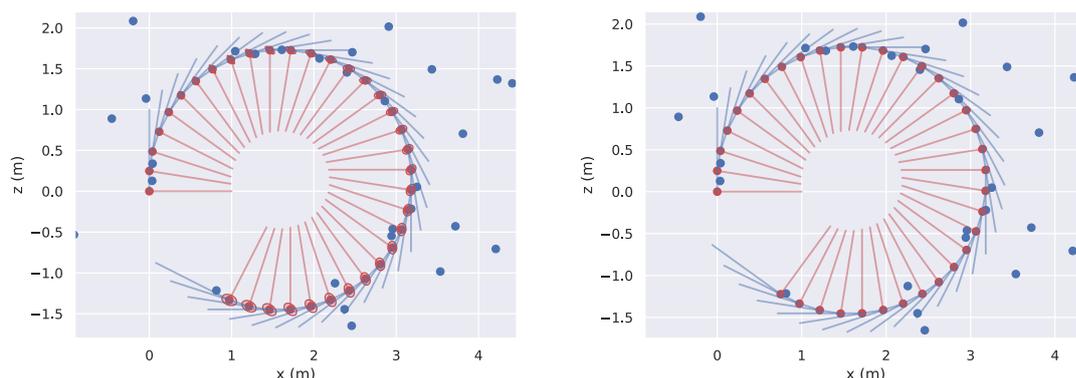


Figura 7.3: Estimaciones de pose y covarianzas correspondientes a los fotogramas 36 (izquierda) y 37 (derecha) de la simulación de la trayectoria circular utilizando GBA. Las líneas azules y rojas representan los ejes x y z de los marcos locales de cada pose. Las poses verdes son las seleccionadas para la ventana de optimización, las rojas son las que quedan fuera. Las elipses representan intervalos de confianza del 99 % para la posición. Los puntos azules son marcas de referencia cercanas.

- CPUs 2Ghz Cortex™ Octa-core
- 2GB LPDDR3 RAM
- 2× puertos USB 3.0
- Un tamaño de 83 × 58 mm.
- Ubuntu 18.04 OS

Como cámara se utilizó una *Tara USB 3.0 Stereo Vision Camera* de *e-con Systems* con las siguientes características:

- interfaz USB 3.0
- fotogramas estéreo de tamaño 640 × 480 @30Hz
- IMU LSM6DS0 de seis ejes integrada

El montaje final de todos los dispositivos en la plataforma hexápoda se puede apreciar en la Figura 7.8.

Para simular la navegación sobre terrenos irregulares en un entorno controlado, se diseñó y montó una pista de prueba experimental dentro de un laboratorio como se muestra en las figuras 7.11 y 7.9. La pista de prueba sigue una trayectoria cuadrada de aproximadamente 8m que involucra obstáculos tales como un conjunto de escaleras, una rampa y una colina de cubos de altura irregular. Cada uno de estos tipos de terreno representa un desafío diferente para el sistema S-PTAM + AMG. La pila de bloques de madera irregulares simula un terreno irregular donde el sistema necesita adaptarse a las pequeñas pero impredecibles diferencias de altura en el entorno. Al subir las escaleras, las diferencias de altura se vuelven más regulares pero también

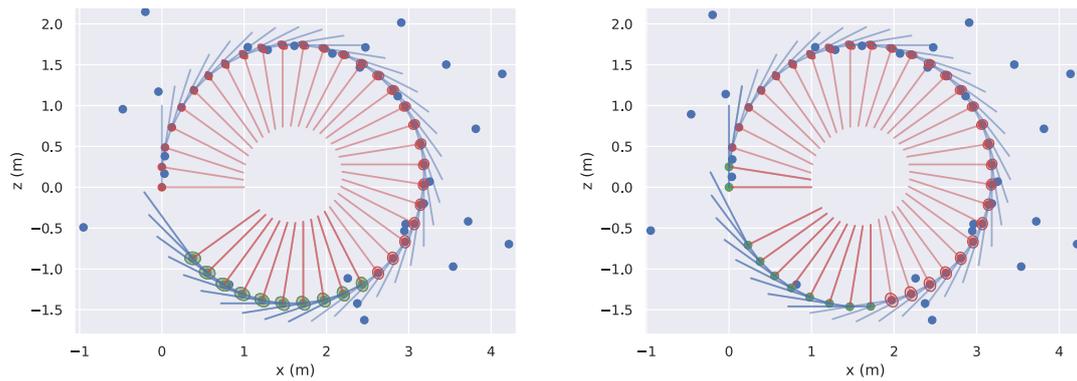


Figura 7.4: Estimaciones de pose y covarianzas correspondientes a los fotogramas 36 (izquierda) y 37 (derecha) de la simulación de la trayectoria circular utilizando una ventana local de 10 fotogramas clave con prior unario. Las líneas azules y rojas representan los ejes x y z de los marcos locales de cada pose. Las poses verdes son las seleccionadas para la ventana de optimización, las rojas son las que quedan fuera. Las elipses representan intervalos de confianza del 99 % para la posición. Los puntos azules son marcas de referencia cercanas.

tienen una altura más desafiante. Al descender la rampa, el sensor visual del robot estará apuntando al suelo, donde solo se puede ver un número limitado de características que no resultan suficientes para realizar una estimación visual robusta. En todos estos casos, el sistema también está sujeto a resbalones en los que incluso el AMG no podrá mantener estable el el cuerpo del robot, y la cámara sufrirá movimientos repentinos e impredecibles. Los giros en cada vértice de la trayectoria cuadrada también representan un desafío adicional para el sistema SLAM debido al desenfoque en las imágenes presente al realizar rotaciones rápidas.

Además, se configuró un sistema de estimación de verdad terrestre para comparar el desempeño del sistema de localización durante cada experimento contra una fuente externa más confiable. Se utilizan dos versiones diferentes de este sistema. Los experimentos más antiguos utilizan una combinación del sistema de seguimiento de etiquetas whycon [92] para la estimación de la posición, y una unidad de medición inercial XSens MTi- 30 que proporciona una estimación de la orientación absoluta filtrada como se muestra en la Figura 7.11. Los experimentos más recientes se basan en el seguimiento de patrones AprilTag [93] por una cámara externa como se muestra en Figura 7.9. El seguimiento de estas etiquetas en cada fotograma proporciona la posición, así como la estimación de orientación con respecto al el marco de referencia de la cámara montada, con una precisión esperada de $\pm 0,03\text{m}$ y $\pm 2\text{deg}$. Dado que ambos sistemas de verdad terrestre son independientes del sistema experimental, las estimaciones de pose deben alinearse temporalmente a mano antes de procesarse.

Aunque el experimento ideal hubiese sido permitir que el robot navegue de forma autónoma sobre una configuración de terreno irregular, algunos problemas técnicos



Figura 7.5: Plataforma *PhantomX Mark II hexapod crawler*.

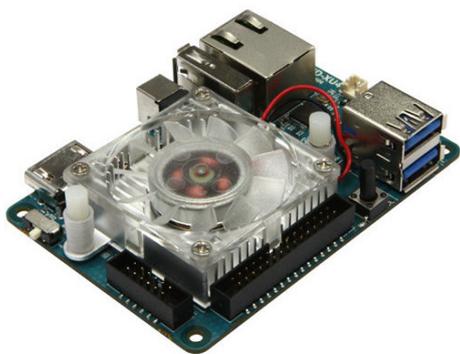


Figura 7.6: Computadora ODROID-XU4. **Figura 7.7:** Cámara estéreo E-con Tara.

que imposibilitaron que el sistema de caminata adaptativo y el sistema S-PTAM se pudiesen ejecutar simultáneamente en la misma plataforma física. Específicamente, el AMG es controlado mediante un bucle muy sensible que espera un rendimiento en tiempo real al nivel de unos pocos milisegundos. Cualquier diferencia en la velocidad de respuesta da como resultado mediciones de error de posición que no escalan con el valor de umbral que se utiliza para verificar las colisiones contra el suelo. Incluso al ejecutar cada proceso en su propio núcleo dedicado, el bus USB utilizado por el sistema AMG para la comunicación sigue siendo un cuello de botella, ya que también está constantemente ocupado con las imágenes que se transmiten desde las cámaras.

Por esta razón, el experimento ideal se divide en dos más pequeños e independientes: por un lado, el sistema de localización se prueba en el contexto de la navegación en terreno irregular, aunque está guiado por un operador humano y no por un sistema de navegación autónoma de lazo cerrado. Por otro lado, la factibilidad del uso del sistema de localización propuesto como un estimador para la navegación autónoma se prueba utilizando un sistema de navegación autónoma de lazo cerrado para guiar

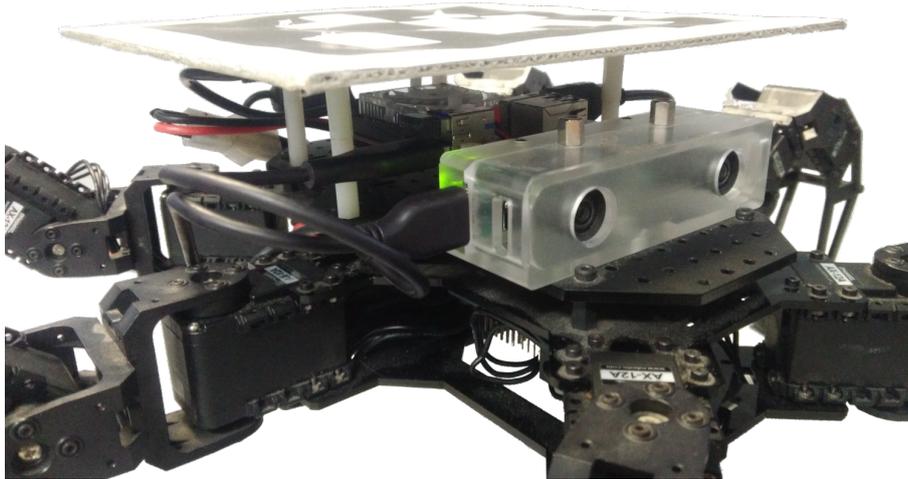


Figura 7.8: Montaje final de la cámara estéreo Tara y la unidad de procesamiento Odroid-XU4 en la plataforma hexápoda PhantomX.

al robot, aunque utilizando un patrón de caminata más simple sin detección de colisión para navegar sobre una superficie lisa.

A continuación, se analizan tres instancias experimentales para resaltar las diferentes propiedades del sistema de localización propuesto. En Sección 7.2.1 se presentan experimentos sobre la idoneidad del sistema SLAM utilizado y la elección de una IMU para hacer frente a los movimientos impredecibles inherentes a las formas de locomoción de robots con patas. Para esto, el robot es guiado por una configuración de terreno irregular por un operador humano mientras el sistema SLAM se ejecuta a bordo, y el rendimiento se compara con otro sistema SLAM del estado del arte que se bajo en las mismas condiciones. Luego, en las subsecciones 7.2.2 y 7.2.3 se presentan los experimentos mencionados en el párrafo anterior, que hacen foco en la navegación sobre terrenos difíciles y la factibilidad de la navegación autónoma respectivamente.

7.2.1. Factibilidad del sistema de SLAM

Los primeros experimentos realizados con el robot hexápodo sobre un terreno irregular se realizaron en una versión antigua del sistema SLAM visual-inercial. Esta versión incluía un filtro EKF como forma de fusionar las medidas visuales e inerciales a modo de bucle cerrado, como se muestra en la Figura 7.10. Este enfoque también se denomina integración *loosely-coupled*, en contraste con las integraciones *tightly-coupled* que integran mediciones inerciales directamente como restricciones en el problema de Bundle Adjustment.

Otra diferencia con la configuración final consistió en utilizar sensores diferentes, particularmente una cámara estéreo Bumblebee 2 y una unidad de medición inercial

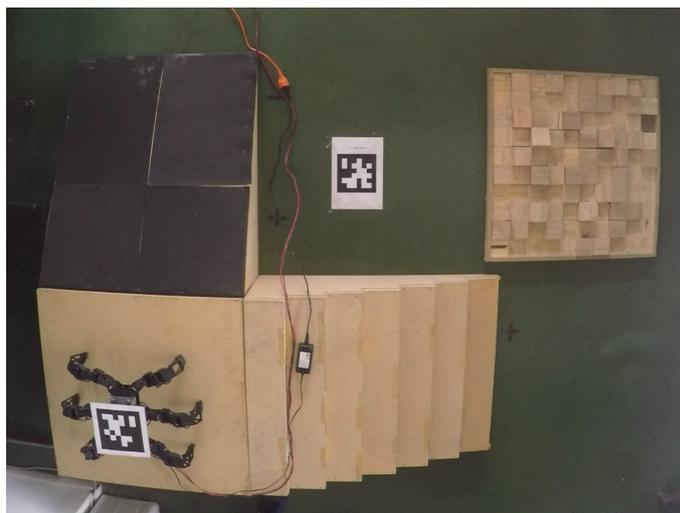


Figura 7.9: La configuración experimental vista desde una cámara montada en el techo utilizada para estimar las ubicaciones del patrón AprilTag.

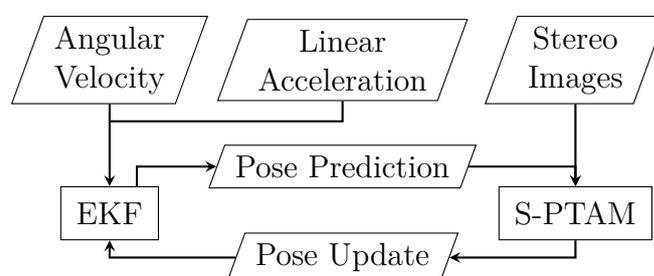


Figura 7.10: Integración débilmente acoplada de S-PTAM e IMU a través de un filtro EKF.

XSens MTi- 30. Más tarde, la cámara Bumblebee fue reemplazada por una Tara (7.7) debido a la interfaz Firewire deprecada de la primera y su gran peso, que provocaba que el robot hexápodo cabeceara fuertemente a cada paso. Dado que la cámara más nueva posee una IMU integrada, la XSens, voluminosa y de alto precio también se descartó.

Durante los experimentos, se realizó una comparación con el sistema popular del estado del arte RGB-D SLAM [94], que utiliza SLRs (cámara estilo Kinect) para evaluar la factibilidad del enfoque estéreo propuesto.

El robot fue guiado a lo largo del circuito por un operador humano mientras se registraban los datos sensoriales. Ambos algoritmos de SLAM se ejecutaron luego en una computadora de escritorio con un procesador Intel® Core™ (TM) i7-4790 CPU @ 3.60GHz, aunque se utilizó el sistema ROS [95] para simular el comportamiento en tiempo real. Desafortunadamente las imágenes de la cámara y los fotogramas RGB-D no pueden grabarse simultáneamente porque el patrón infrarrojo proyectado por el sensor RGB-D interfiere con el sensor de imagen de la cámara Bumblebee, lo cuál

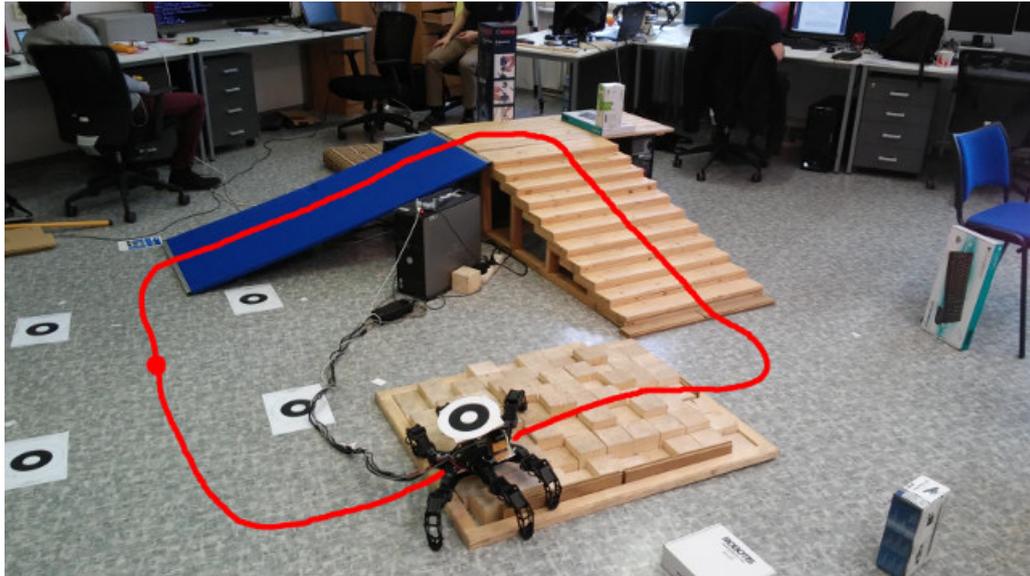


Figura 7.11: Configuración experimental de la pista para simular terrenos irregulares.

causa un patrón de ruido constante.

El sistema de localización externo WhyCon [92] se usó para proporcionar datos fiables de verdad terrestre sobre la trayectoria realizada. Los errores de estimación de posición suelen estar en el rango de 1 cm por eje. WhyCon se diseñó originalmente para proporcionar una posición 2D, y la precisión de la estimación 3D depende en gran medida de la resolución de la cámara y del tamaño del patrón. Esto se vuelve un problema especialmente cuando se intenta medir la orientación 3D. Aprovechando la orientación absoluta y confiable dada por la IMU XSens MTi-30, se consideran estas mediciones en lugar de las calculadas por WhyCon para los datos de orientación de verdad terrestre.

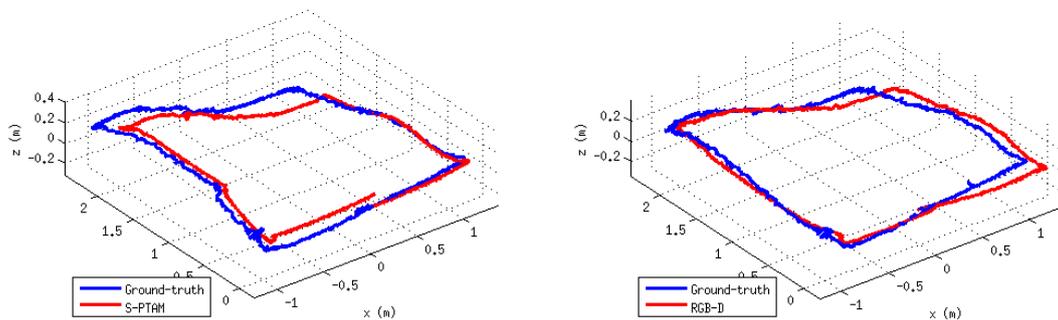


Figura 7.12: Trayectorias estimadas respecto de las mediciones de verdad terrestre.

En la Figura 7.12 dos trayectorias estimadas por S-PTAM y RGB-D SLAM pueden verse representadas contra sus respectivas verdades terrestres. Ambas trayectorias muestran tener un buen desempeño y, a simple vista, RGB-D parece superar al S-PTAM en términos de precisión.

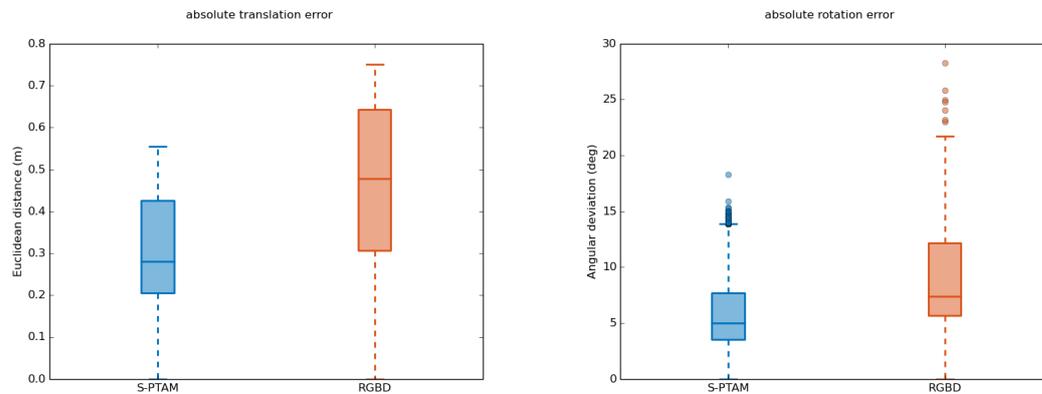


Figura 7.13: Errores de traslación y rotación absolutos para las estimaciones de pose realizadas por ambos algoritmos SLAM con respecto al sistema de verdad terrestre WhyCon.

Los errores calculados para las trayectorias estimadas por ambos métodos según las métricas planteadas en la Sección 3.5 se pueden apreciar en las figuras 7.13 y 7.14. Podemos observar un rendimiento ligeramente mejor de SPTAM al comparar la métrica del error relativo, aunque el RGB-D supera al S-PTAM en términos del error absoluto.

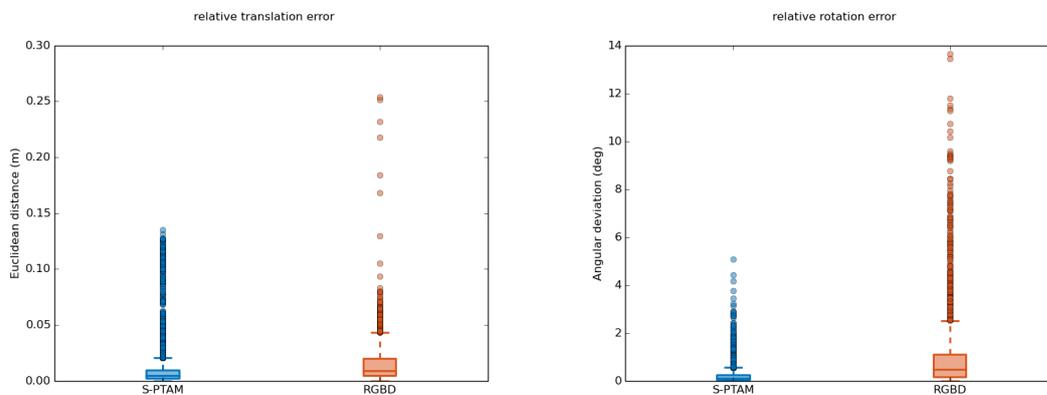


Figura 7.14: Errores de traslación y rotación relativos para las estimaciones de pose realizadas por ambos algoritmos de SLAM con respecto al sistema de verdad terrestre WhyCon.

En términos de desempeño en tiempo real del sistema de localización, S-PTAM superó a la SLAM RGB-D brindando una estimación de localización a 10 Hz sobre 3 a 4 Hz del último.

7.2.2. Navegación en terrenos irregulares

Para demostrar la factibilidad del sistema SLAM propuesto para la navegación en terrenos irregulares utilizando robots hexápodos, un operador humano guía al robot en una configuración de terreno accidentado mientras el sistema SLAM se ejecuta a bordo. Al igual que la pista utilizada en los primeros experimentos, la navegación sigue una trayectoria cuadrada de aproximadamente 8m que involucra obstáculos tales como un conjunto de escaleras, una rampa y una colina de cubos de altura irregular. La configuración se muestra, desde el punto de vista de la cámara que registra los datos de verdad terrestre, en la Figura 7.9.

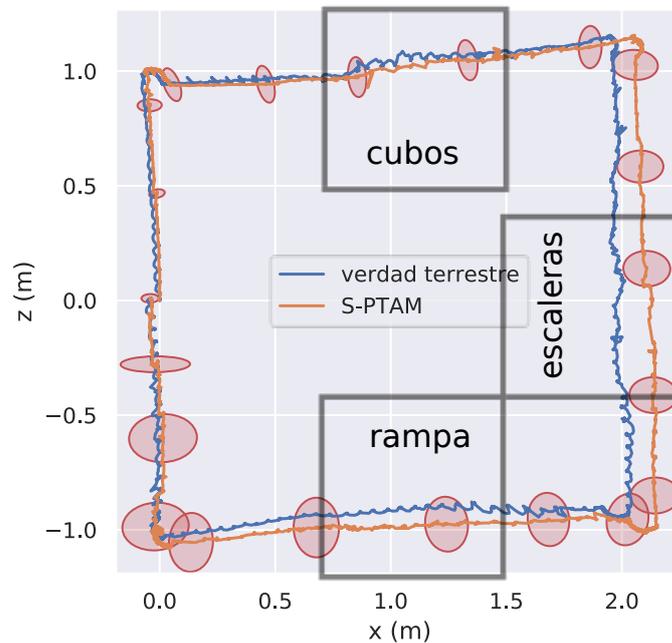


Figura 7.15: Trayectorias estimadas por el sistema S-PTAM durante la navegación en terreno irregular y verdad terrestre, según lo registrado por el localizador de AprilTag, como referencia. Las elipses denotan intervalos de confianza de posición estimados (95 %) en puntos específicos de la trayectoria.

La Figura 7.15 muestra tanto la trayectoria estimada por el sistema SLAM como la estimada por el sistema de verdad terrestre, así como los obstáculos atravesados dibujados sobre el fondo. Ambas trayectorias representan el camino recorrido por el marco de referencia de la cámara izquierda con respecto a su primer pose, es decir, el origen euclidiano en el que están representados todos los elementos del mapa. Se puede ver que la trayectoria estimada por S-PTAM se desvía de la verdad terrestre, pero una vez que se cierra un bucle cuando se regresa cerca del origen, se corrige el error de deriva acumulado. La Figura 7.16 muestra la evolución de ambas trayectorias a lo largo del tiempo para cada parámetro de pose de forma independiente, donde se

puede ver cómo el sistema S-PTAM subestima en gran medida el cambio de altura al subir el conjunto de escaleras y también al bajar por la rampa.

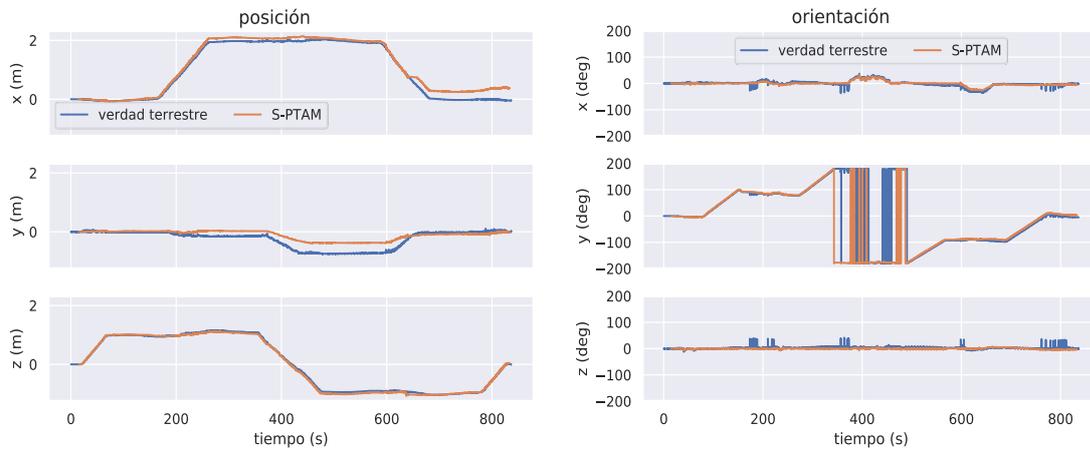


Figura 7.16: Trayectoria estimada por el sistema S-PTAM (naranja) durante la navegación sobre terreno irregular y verdad terrestre, según lo registrado por el localizador de AprilTag (azul), como referencia.

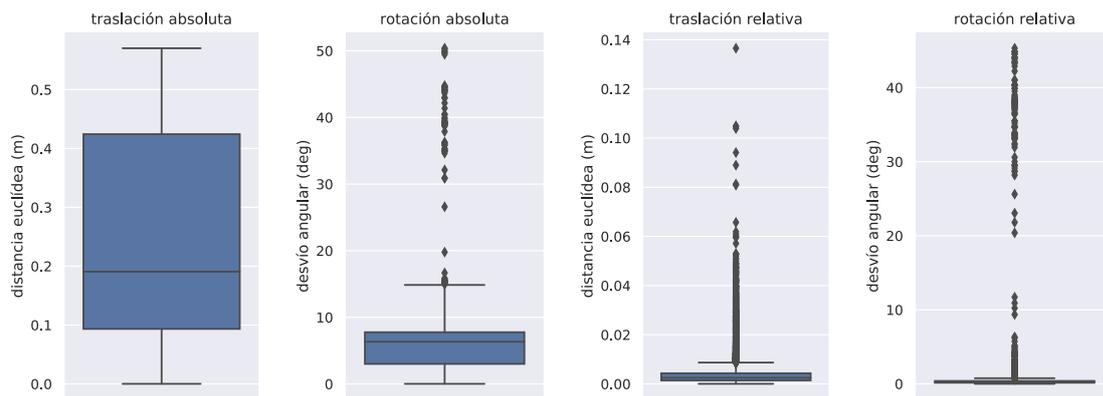


Figura 7.17: Estadísticas de error de la trayectoria estimada contra la verdad terrestre para el experimento de navegación sobre terreno accidentado.

En cuanto a la estimación de incertidumbre, las figuras Figura 7.18 y Figura 7.19 muestran la evolución del estimador de covarianza durante el experimento de navegación sobre terrenos irregulares. Las áreas rojas indican intervalos de tiempo en los que el robot está caminando sobre un obstáculo, específicamente, y en orden cronológico: el conjunto de cubos irregulares, las escaleras y la rampa de bajada. Las áreas amarillas corresponden a los momentos en que el robot está girando. Se pueden discernir algunos artefactos interesantes que ameritan algunos comentarios.

Alrededor de la marca de tiempo 650s se puede ver un pico en la Figura 7.18 que afecta principalmente a la posición, y que al analizar las estimaciones en las dimensiones

individuales (Figura 7.19) se puede atribuir a un crecimiento de la incertidumbre a lo largo del eje z del robot. Esto corresponde al momento en que el robot está bajando por la rampa, donde las cámaras solo logran ver el suelo plano que no posee muchos rasgos distintivos. Por lo tanto la estimación de la pose se realiza mediante el seguimiento de un número muy bajo de características de la imagen, o únicamente mediante el modelo de predicción, donde la incertidumbre asociada también es alta. Sin embargo, el algoritmo se recupera luego de volver a observar una parte del mapa más rica en características, y las estimaciones de incertidumbre regresan al nivel anterior.

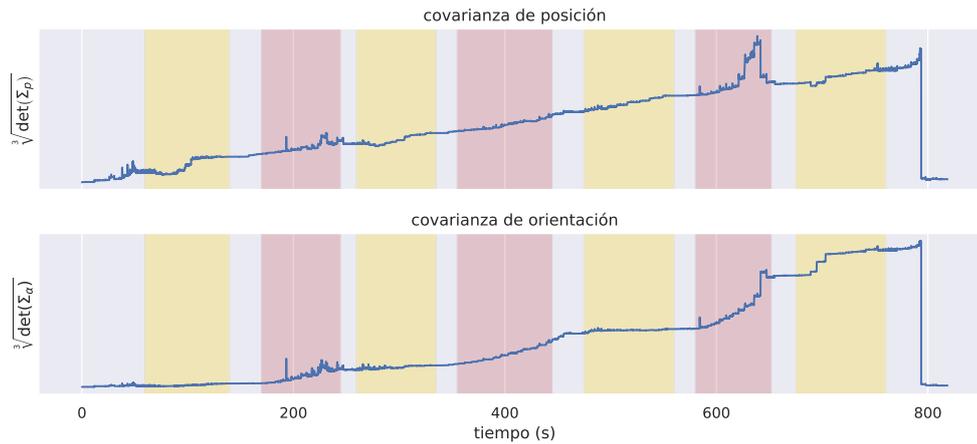


Figura 7.18: Evolución del volumen de covarianza de la posición y rotación de la cámara en la trayectoria del experimento sobre navegación de terreno accidentado.

Como es de esperar, las estimaciones de covarianza disminuyen en todas las dimensiones luego de que se vuelve a observar el origen de la secuencia y se cierre un ciclo. Como ya fue visto durante los experimentos sobre escenarios simulados, una vez que cualquiera de los fotogramas clave iniciales comparte un alto grado de co-visibilidad con cualquiera de los parámetros en la ventana activa, se seleccionará como referencia para la estimación de la incertidumbre de la misma.

Para evaluar el rendimiento en tiempo real del sistema de navegación autónomo implementado, los perfiles de las operaciones temporalmente más costosas se registraron mientras se ejecutaba el sistema en la computadora a bordo Odroid. Los resultados del módulo de seguimiento en tiempo real se pueden observar en la Figura 7.20. Se puede ver que, en promedio, se tarda unos 0,2 a 0,25 segundos para manejar un fotograma entrante y proporcionar una estimación de pose. Aunque esto es suficiente para controlar una plataforma hexápoda, puede resultar problemático para vehículos con una dinámica más veloz, p.e. robots multicopteros. También se puede ver que la operación que consume más tiempo es el proceso de extracción de características, que podría evitarse, por ejemplo mediante el uso de extractores más rápidos o un módulo de hardware dedicado.

Por completitud, el perfil para la rutina de optimización del mapa se muestra en

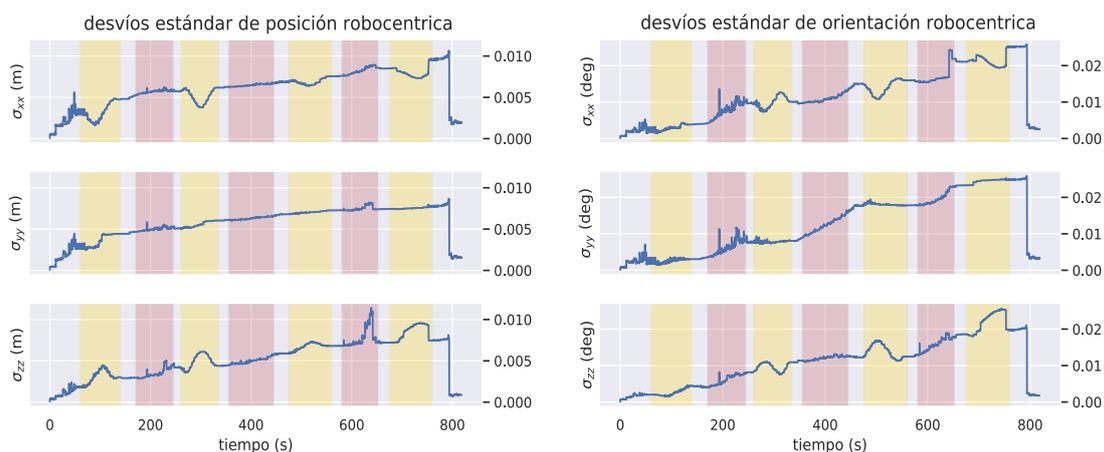


Figura 7.19: Evolución de los desvíos estándar en ejes individuales de la posición y rotación de la cámara en la trayectoria del experimento sobre navegación de terreno accidentado.

Figura 7.21. Aunque parece no depender del tamaño del mapa, como es esperado, el proceso sigue siendo bastante costoso. Sin embargo, los fotogramas clave entrantes son procesados continuamente sin acumularse, por lo tanto el elevado costo computacional no limita el correcto funcionamiento del sistema.

7.2.3. Navegación autónoma

Para demostrar la factibilidad del sistema S-PTAM para la navegación autónoma, se instruyó al robot navegar por un escenario simple de $1,5m \times 1m$ sobre un piso plano. Para ello, el robot se basó únicamente en la localización autónoma provista por el sistema de SLAM hasta alcanzar nuevamente la posición de salida. El escenario está libre de obstáculos ya que, como fue mencionado anteriormente, dificultades técnicas impiden que el sistema SLAM se ejecute a la par del algoritmo Adaptive Motion Gait en la configuración de hardware actual. Por lo tanto, se utilizó la misma marcha del trípode, aunque sin detección de colisión contra el suelo.

La trayectoria mostrada en la Figura 7.22, aunque errática en momentos específicos, parece funcionar lo suficientemente bien como para regresar al origen incluso sin la ayuda del módulo de cierre de ciclos. Al analizar la trayectoria sobre parámetros de pose individuales (Figura 7.23), se puede ver que alrededor de la marca de 250s hay una medida errática en el eje vertical (y) del robot. Esto corresponde a la estimación errática observada en la esquina inferior derecha en la Figura 7.22, probablemente debido a la presencia de errores de asociación. Las estadísticas de error correspondientes a este experimento se muestran en Figura 7.24.

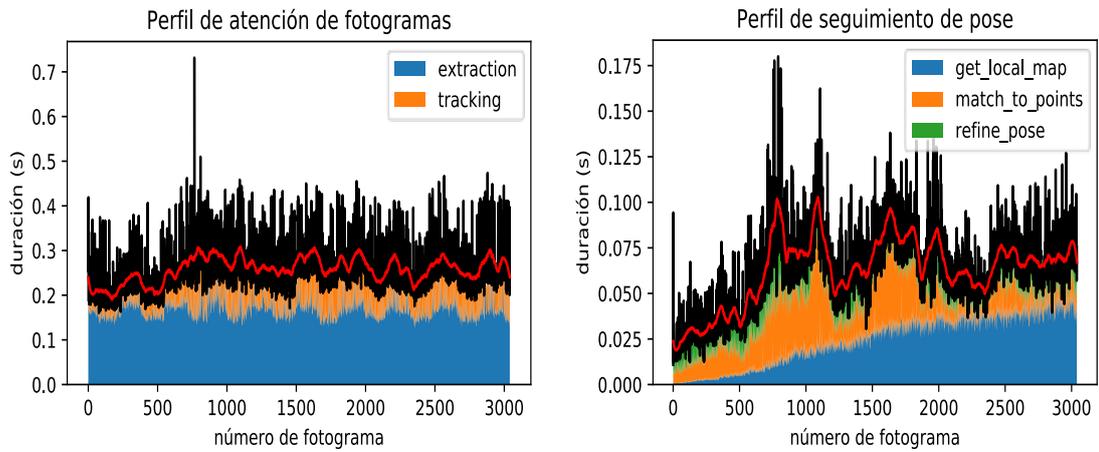


Figura 7.20: Perfiles de rendimiento para la rutina de atención de fotogramas y un detalle de la subrutina de seguimiento de pose. La línea roja representa el tiempo total a través de un filtro pasa bajos (Savitzky-Golay con tamaño de ventana 101, y orden polinomial 3).

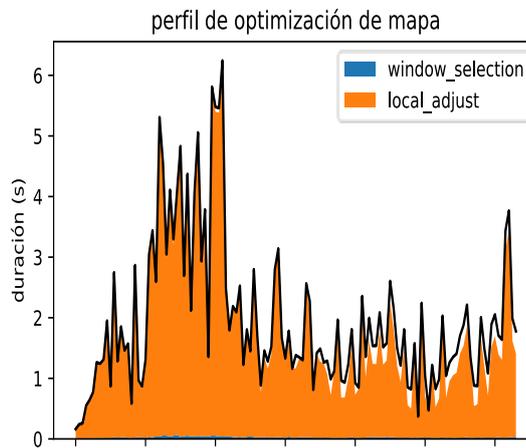


Figura 7.21: Perfiles de rendimiento para la rutina de optimización de mapas.

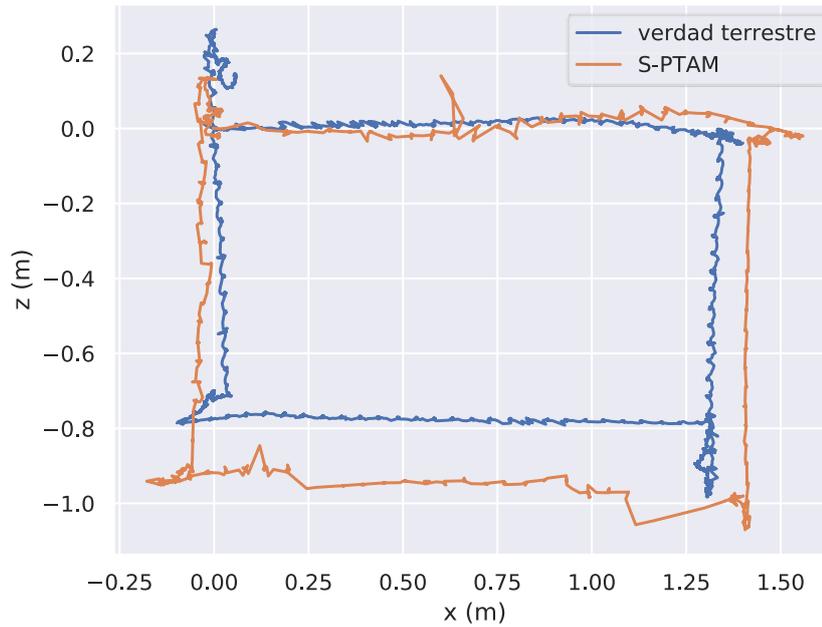


Figura 7.22: Trayectoria estimada por el sistema S-PTAM durante la navegación autónoma (naranja) y la verdad terrestre (azul), según lo registrado por el localizador de AprilTag, como referencia.

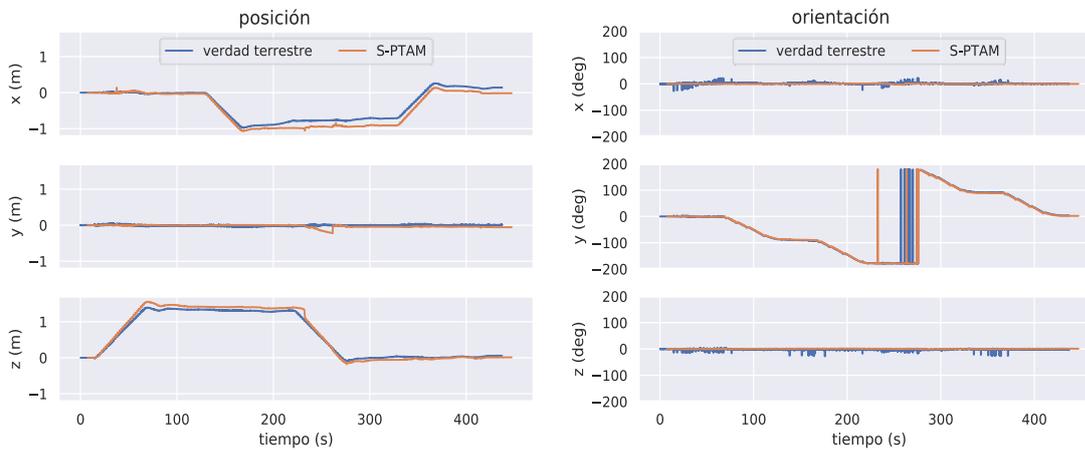


Figura 7.23: Trayectoria estimada en cada eje por el sistema S-PTAM durante la navegación autónoma (naranja) y la verdad terrestre (azul), según lo registrado por el localizador de AprilTag, como referencia.

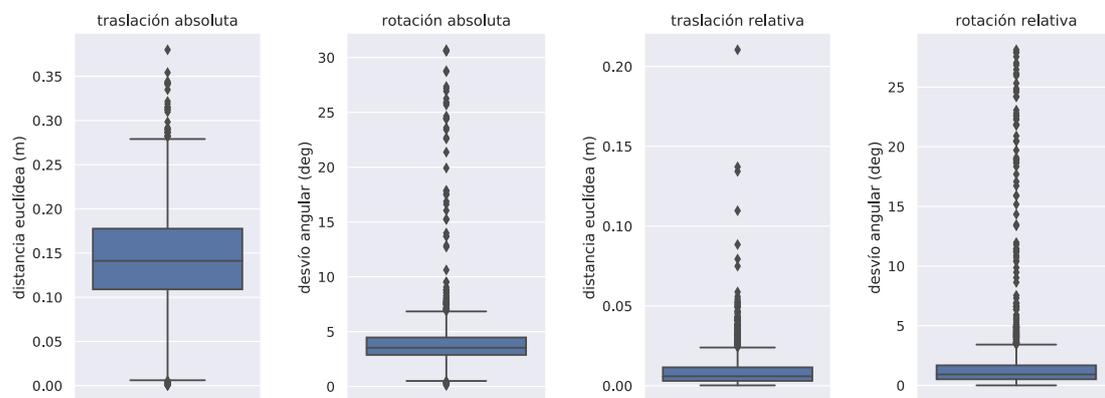


Figura 7.24: Estadísticas de error de la trayectoria estimada en función de la verdad terrestre para el experimento de navegación autónoma.

8 Conclusiones

En esta tesis, se propone un sistema de navegación que permite a un robot hexápodo realizar tareas de navegación autónoma de manera robusta, incluso cuando opera en terrenos difíciles. El sistema se basa en cámaras estereoscópicas para realizar SLAM en tiempo real sobre una computadora a bordo, y proporcionar una estimación de localización fiable con una incertidumbre asociada. Para hacer frente a los movimientos repentinos e impredecibles inherentes a la locomoción de los vehículos con patas en terrenos irregulares, se utiliza una unidad de medición inercial para proporcionar predicciones de rotación precisas entre pares de fotogramas estéreo. Esto reduce el área en la que se deben buscar correspondencias entre características locales de la imagen y las marcas de referencia del mapa, y dificulta que la rutina de optimización de poses caiga en mínimos locales al proporcionar una solución inicial que debería ser cercana al óptimo.

Para permitir que el sistema pueda ejecutarse en tiempo real incluso durante la navegación a largo plazo, se propone una estrategia de selección de ventana para considerar solo una parte delimitada del mapa durante la optimización del mismo. Para habilitar la selección de una ventana de este tipo en complejidad de tiempo acotada, se implementó una estructura de datos dedicada denominada grafo de visibilidad para contener los elementos del mapa y sus relaciones, así como cierta información relacional agregada que permite consultar rápidamente los fotogramas clave cercanos que comparten un alto grado de información. Esta estrategia fue diseñada específicamente para proporcionar estimaciones de los parámetros en forma de variables aleatorias con distribuciones normales multivariadas. Esto significa que además de proporcionar estimaciones numéricas para las poses de la cámara relacionadas a cada fotograma estéreo entrante y las posiciones de marcas de referencia mapeadas, también se proporcionan estimaciones de incertidumbre en forma de matrices de varianza-covarianza con la dimensionalidad correspondiente. La solución canónica para una ventana probabilísticamente consistente es la marginalización adecuada de las variables que quedan fuera, que tiene una complejidad que crece con el tamaño del mapa. Para hacer frente a este problema, se propone una heurística novedosa para aproximar la verdadera marginal en forma de un prior unario sobre uno de los parámetros seleccionados de la ventana. Se muestra que la estrategia se comporta de la manera esperada en una serie de experimentos en entornos de simulación controlados.

Finalmente, una serie de experimentos realizados en una plataforma hexápoda real muestra la factibilidad del sistema de navegación desarrollado para la tarea prevista.

Los mismos son realizados sobre una pista de prueba con obstáculos específicamente diseñada. A pesar de que los algoritmos AMG y S-PTAM no pudieron ejecutarse simultáneamente en la misma plataforma debido a limitaciones de hardware, todos los aspectos del sistema propuesto se probaron en dos experimentos independientes. Por un lado, el robot fue capaz de navegar continuamente sobre los obstáculos a la vez que proporcionaba una localización confiable en tiempo real con todos los sistemas ejecutados en una computadora de a bordo. Por otro lado, se realizó un experimento de navegación autónoma en el que el robot siguió una ruta propuesta sin la necesidad de un operador humano, aunque sin tener que superar obstáculos en el camino.

A pesar de haber superado con éxito el problema inicialmente propuesto de navegar por terrenos irregulares con una plataforma hexápodo con alto grado de autonomía, se encontraron algunas limitaciones en el diseño que pueden desencadenar interesantes temas de investigación en el futuro y proporcionar motivación para continuar con esta línea de trabajo. Además, durante el tiempo en que se estaba desarrollando el trabajo actual, surgieron nuevos enfoques y algoritmos que arrojaron una nueva luz sobre algunos de los problemas que aquí se analizan y que merecen un análisis más detallado. En este espíritu, los siguientes párrafos proponen algunas posibles líneas de investigación que se basan en el trabajo que ya fue realizado en esta tesis.

8.1. Trabajo futuro

La estrategia de prior unario presentada y utilizada a lo largo de este trabajo presenta el problema de la restricción insuficiente de los parámetros de la frontera de la ventana con respecto al problema global. Debido a que los parámetros de frontera en el contexto de la ventana de visibilidad compartida implementada son esencialmente marcas de referencia, quedan sub-representadas solo en el caso de que una única medida monocular permanezca dentro de la ventana o si el ángulo máximo de paralaje entre las medidas existentes es demasiado bajo. Dado que este escenario es extremadamente raro y fácil de evitar, se ignoró en los experimentos realizados. No obstante, la solución puede diferir de la global, ya que no se tienen en cuenta algunas restricciones. Una solución sencilla puede ser modelar las posiciones de marcas de referencia sin restricciones utilizando la parametrización de profundidad inversa [96]. Un análisis más detallado de la frontera también puede revelar una mejor manera de formular el anterior, posiblemente en forma de una restricción más general sobre todas las variables en la frontera. Si bien la estructura del anterior sería la misma que la de la marginal verdadera, no resulta sencillo [83] elegir valores para un prior que puedan calcularse en tiempo constante y sean una buena aproximación de la verdadera marginal.

Además, trabajos recientes [11, 79, 59] parecen estar enfocados en el llamado SLAM visual-inercial estrechamente acoplado. Al agregar mediciones de IMU preintegradas al problema de Bundle Adjustment como un tipo nuevo de factor, los sesgos y la

dirección del vector de gravedad se pueden estimar como parámetros para cada fotograma. Esto permite separar las mediciones de aceleración lineal de la fuerza de gravedad e integrarlas para calcular una estimación de velocidad lineal, que a su vez puede usarse para alimentar el modelo predictivo con datos más precisos que el modelo de velocidad constante empleado en el trabajo actual.

También sería interesante explorar las posibilidades de trabajar con transformaciones relativas [97] en lugar de absolutas, y los inconvenientes introducidos por mayores dependencias condicionales durante la optimización. Las incertidumbres relativas podrían entonces integrarse bajo demanda y utilizarse para realizar una asociación activa. Algunos problemas de concurrencia también pueden resolverse de esta manera, ya que las actualizaciones de parámetros introducidas mediante la ejecución simultánea de subprocesos, como cierres de ciclo, no necesitarían ser detectadas y administradas específicamente. Además, puede ser posible que se puedan usar relaciones relativas para des-marginalizar parámetros de manera más eficiente durante los cierres de ciclo grandes.

A nivel de programación, se observaron casos en los que un parámetro podría quedar sub-representado debido a la eliminación de las restricciones asociadas por ser marcadas como valores atípicos. El minimizador ceres utilizado como módulo de optimización, aunque permite calcular distribuciones marginales de parámetros, no los calcula aprovechando la distribución esparsa de los bloques de parámetros en la matriz de información, sino que utiliza un algoritmo de factorización QR para calcular la matriz de covarianza del problema completo como paso preliminar. Este algoritmo, al contrario que el algoritmo marginal de bloques esparsos [98], no admite parámetros sin restricciones y, por lo tanto, falla en el caso mencionado.

A nivel de hardware, sería interesante separar los procesos sensibles al tiempo, en particular el controlador de movimiento y el controlador de la cámara, ya sea en una distribución de Linux en tiempo real o en unidades de control dedicadas. Varios problemas, p.e. los malfuncionamientos de controladores, se relacionaron con el desborde de los canales de comunicación debido a que, de a momentos, todos los recursos está dedicados a resolver problemas computacionalmente complejos, como la eventual rutina de Bundle Adjustment cuando se inserta un fotograma clave.

Bibliografía

- [1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.
- [2] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras,” *CoRR*, vol. abs/1610.06475, 2016. [Online]. Available: <http://arxiv.org/abs/1610.06475>
- [3] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civera, and J. J. Berles, “S-ptam: Stereo parallel tracking and mapping,” *Robotics and Autonomous Systems*, vol. 93, pp. 27–42, 2017.
- [4] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, “Keyframe-Based Visual-Inertial SLAM Using Nonlinear Optimization,” *Proc. of Robot.: Sci. and Syst.*, p. 0, 2013.
- [5] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “Monoslam: Real-time single camera slam,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.
- [6] L. Clement, J. Kelly, and T. D. Barfoot, “Robust monocular visual teach and repeat aided by local ground planarity and color-constant imagery,” *Journal of Field Robotics*, vol. 34, no. 1, pp. 74–97, 2017.
- [7] B. Steder, G. Grisetti, C. Stachniss, and W. Burgard, “Visual slam for flying vehicles,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1088–1093, Oct 2008.
- [8] A. Schmidt and A. Kasiński, “The visual slam system for a hexapod robot,” *Computer vision and graphics*, pp. 260–267, 2010.
- [9] F. Hidalgo and T. Bräunl, “Review of underwater slam techniques,” in *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, Feb 2015, pp. 306–311.
- [10] M. W. Achtelik, S. Lynen, S. Weiss, L. Kneip, M. Chli, and R. Siegwart, “Visual-inertial slam for a small helicopter in large outdoor environments,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 2651–2652.
- [11] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe-based visual-inertial odometry using nonlinear optimization,” *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015. [Online]. Available: <https://doi.org/10.1177/0278364914554813>

-
- [12] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, Nov. 2007, pp. 1–10.
- [13] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based mav navigation in unknown and unstructured environments," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 21–28.
- [14] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments," *J. Field Robot.*, vol. 28, no. 6, pp. 854–874, Nov. 2011. [Online]. Available: <http://dx.doi.org/10.1002/rob.20412>
- [15] C. Mei, G. Sibley, and P. Newman, "Closing loops without places," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 3738–3744.
- [16] F. Dellaert and M. Kaess, "Square root sam: Simultaneous localization and mapping via square root information smoothing," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.
- [17] M. H. Raibert, H. B. Brown, and S. S. Murthy, "Machines that walk," in *Robotics and Artificial Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, pp. 345–364.
- [18] M. Kaneko, M. Abe, and K. Tanie, "A hexapod walking machine with decoupled freedoms," *IEEE Journal on Robotics and Automation*, vol. 1, no. 4, pp. 183–190, Dec. 1985.
- [19] M. Kaneko, K. Tanie, and M. N. M. Than, "A control algorithm for hexapod walking machine over soft ground," *IEEE Journal on Robotics and Automation*, vol. 4, no. 3, pp. 294–302, Jun. 1988.
- [20] N. Porcino, "Hexapod gait control by a neural network," *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, vol. 1, pp. 189 – 194, 1990.
- [21] G. Parker and G. Rawlins, "Cyclic genetic algorithms for the locomotion of hexapod robots," *Proceedings of the World Automation Congress (. . . , 1996*.
- [22] M. A. Jimenez and P. Gonzalez De Santos, "Attitude and position control method for realistic legged vehicles," *Robotics and Autonomous Systems*, vol. 18, no. 3 SPEC. ISS., pp. 345–354, Aug. 1996.
- [23] M. Antonia Jiménez and P. González de Santos, "Attitude and position control for non-rigid walking machines," *Mechanism and Machine Theory*, vol. 33, no. 7, pp. 1013–1029, Oct. 1998.
- [24] J. Mrva and J. Faigl, "Tactile sensing with servo drives feedback only for blind hexapod walking robot," in *2015 10th International Workshop on Robot Motion and Control, RoMoCo 2015*, 2015, pp. 240–245.

- [25] B. Gassmann, F. Zacharias, J. M. Zollner, and R. Dillmann, "Localization of walking robots," in *ICRA*, 2005, pp. 1471–1476.
- [26] A. J. Davison, "Real-time Simultaneous Localisation and Mapping with a Single Camera," *Iccv*, vol. 2, pp. 1403–1410, 2003. [Online]. Available: <http://ieeexplore.ieee.org/search/selected.jsp>
- [27] O. Stasse, A. J. Davison, R. Sellaouti, K. Yokoi, and C. Aist, "Real-time 3D SLAM for Humanoid Robot considering Pattern Generator Information National Institute of Advanced Industrial and Science Technology," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 348–355, 2006. [Online]. Available: https://www.doc.ic.ac.uk/~ajd/Publications/stasse_etal_iros2006.pdf
- [28] D. Belter, K. Walas, and P. Skrzypczyński, "Autonomous hexapod robot - development of an improved mechanical design and control system," *Pomiary Automatyka Robotyka*, vol. 2, pp. 249–258, 2009.
- [29] A. Chilian and H. Hirschmuller, "Stereo camera based navigation of mobile robots on rough terrain," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4571–4576, 2009.
- [30] A. Stelzer, H. Hirschmüller, and M. Görner, "Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain," *International Journal of Robotics Research*, vol. 31, no. 4, pp. 381–402, 2012.
- [31] D. Belter and P. Skrzypczyński, "Precise self-localization of a walking robot on rough terrain using ptam," in *Adaptive Mobile Robotics*. World Scientific, 2012, pp. 89–96.
- [32] D. Belter and P. Skrzypczyński, "Precise self-localization of a walking robot on rough terrain using parallel tracking and mapping," *Industrial Robot: An International Journal*, vol. 40, no. 3, pp. 229–237, Apr. 2013.
- [33] G. Klein, "Visual tracking for augmented reality," Ph.D. dissertation, University of Cambridge, 2006.
- [34] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 1–10. [Online]. Available: <http://dx.doi.org/10.1109/ISMAR.2007.4538852>
- [35] K. Walas and D. Belter, "Messor - versatile walking robot for search and rescue missions," *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. Vol. 5, No. 2, pp. 28–34, 2011.
- [36] M. Bjelonic, T. Homberger, N. Kottege, P. Borges, M. Chli, and P. Beckerle, "Autonomous navigation of hexapod robots with vision-based controller adaptation," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5561–5568, 2017.

-
- [37] R. C. Smith and P. Cheeseman, “On the representation and estimation of spatial uncertainty,” *The International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.
- [38] H. F. Durrant-Whyte, “Uncertain geometry in robotics,” *IEEE Journal on Robotics and Automation*, vol. 4, no. 1, pp. 23–31, 1988.
- [39] R. Smith, M. Self, and P. Cheeseman, “Estimating uncertain spatial relationships in robotics,” in *Autonomous Robot Vehicles*. New York, NY: Springer New York, Mar. 1990, pp. 167–193.
- [40] P. Moutarlier and R. Chatila, “An experimental system for incremental environment modelling by an autonomous mobile robot,” *Experimental Robotics I*, vol. 1560, pp. 327–346, 1989.
- [41] J. Leonard and H. Durrant-Whyte, “Simultaneous map building and localization for an autonomous mobile robot,” *Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, no. 91, pp. 1442–1447, 1991.
- [42] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “Fastslam: A factored solution to the simultaneous localization and mapping problem,” in *Proceedings of the Eighteenth National Conference on Artificial Intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 2002, pp. 593–598.
- [43] M. Montemerlo, S. Thrun, D. Roller, and B. Wegbreit, “Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges,” in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, ser. IJCAI'03. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 1151–1156.
- [44] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A. Y. Ng, “Simultaneous mapping and localization with sparse extended information filters: Theory and initial results,” CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, Tech. Rep., 2004.
- [45] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, “Simultaneous localization and mapping with sparse extended information filters,” *International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.
- [46] T. B. Schön and F. Lindsten, “Manipulating the multivariate gaussian density,” *Div. Automat. Control, Linköping Univ., Linköping, Sweden, Tech. Rep.*, vol. 2, no. 3, pp. 1–10, 2011.
- [47] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment—a modern synthesis,” in *Vision algorithms: theory and practice*. Springer, 1999, pp. 298–372.
- [48] S. J. Julier and J. K. Uhlmann, “A counter example to the theory of simultaneous localization and map building,” in *Proceedings 2001 ICRA. IEEE Inter-*

- national Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 4, May 2001, pp. 4238–4243 vol.4.
- [49] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, “Real time localization and 3d reconstruction,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2006, pp. 363–370.
- [50] A. Eudes and M. Lhuillier, “Error propagations for local bundle adjustment,” *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, vol. 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2411–2418, 2009.
- [51] K. Konolige, “Slam via variable reduction from constraint maps,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, vol. 2005, no. April. IEEE, 2005, pp. 667–672.
- [52] J.-S. Gutmann and K. Konolige, “Incremental mapping of large cyclic environments,” in *Computational Intelligence in Robotics and Automation, 1999. CIRA’99. Proceedings. 1999 IEEE International Symposium on*. IEEE, 1999, pp. 318–325.
- [53] K. Konolige, “Large-scale map-making,” in *AAAI*, 2004, pp. 457–463.
- [54] G. Sibley, “A sliding window filter for slam,” *Technical Report No. CRES-06-004*, pp. 1–17, 2006.
- [55] G. Sibley, L. Matthies, and G. Sukhatme, “A sliding window filter for incremental slam,” *Lecture Notes in Electrical Engineering*, vol. 8 LNEE, pp. 103–112, 2008.
- [56] —, “Sliding window filter with application to planetary landing,” *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608, Jul. 2010.
- [57] E. Eade, P. Fong, and M. E. Munich, “Monocular graph slam with complexity reduction,” *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pp. 3017–3024, 2010.
- [58] L. Carlone, Z. Kira, C. Beall, V. Indelman, and F. Dellaert, “Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4290–4297, 2014.
- [59] N. Keivan, A. Patron-Perez, and G. Sibley, “Asynchronous adaptive conditioning for visual-inertial slam,” *Springer Tracts in Advanced Robotics*, vol. 109, pp. 309–321, 2016.
- [60] D. Kopitkov and V. Indelman, “No belief propagation required: Belief space planning in high-dimensional state spaces via factor graphs, the matrix determinant lemma, and re-use of calculation,” *International Journal of Robotics Research*, vol. 36, no. 10, pp. 1088–1130, 2017.

-
- [61] R. A. Johnson, “Wichern. dw applied multivariate statistical analysis,” *New Jersey: Prentice-Hall, Inc*, vol. 1, p. 998, 2007.
- [62] J. L. Blanco, J. González-Jiménez, and J. A. Fernández-Madrigal, “An alternative to the mahalanobis distance for determining optimal correspondences in data association,” *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 980–986, 2012.
- [63] T. D. Barfoot and P. T. Furgale, “Associating uncertainty with three-dimensional poses for use in estimation problems,” *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.
- [64] E. Eade, “Lie groups for 2d and 3d transformations,” *URL <http://ethaneade.com/lie.pdf>, revised Dec*, pp. 1–24, 2013.
- [65] H. Strasdat, “Sophus – c++ implementation of lie groups using eigen,” <https://github.com/strasdat/Sophus>, 2017.
- [66] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, “On measuring the accuracy of SLAM algorithms,” *Autonomous Robots*, vol. 27, no. 4, pp. 387–407, 2009. [Online]. Available: <http://dx.doi.org/10.1007/s10514-009-9155-6>
- [67] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision Meets Robotics: The KITTI Dataset,” *The International Journal of Robotics Research, IJRR*, vol. 32, no. 11, pp. 1231–1237, September 2013. [Online]. Available: <http://dx.doi.org/10.1177/0278364913491297>
- [68] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
- [69] F. Dellaert and M. Kaess, “Factor graphs for robot perception,” *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017. [Online]. Available: <http://dx.doi.org/10.1561/23000000043>
- [70] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.
- [71] D. Koller, N. Friedman, and F. Bach, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [72] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [73] D. C. Brown, “A solution to the general problem of multiple station analytical stereo triangulation,” Technical Report RCA-MTP Data Reduction, Patrick Airforce Base, Florida, Tech. Rep. Technical Report No. 43 (or AFMTC TR 58-8), 1958.
- [74] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.

- [75] L. Matthies and S. Shafer, “Error modeling in stereo navigation,” *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 239–248, June 1987.
- [76] M. Paton, K. Mactavish, M. Warren, and T. D. Barfoot, “Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat,” *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016–Novem, pp. 1918–1925, 2016.
- [77] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: A versatile and accurate monocular slam system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
- [78] Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang, and S. Shen, “Autonomous aerial navigation using monocular visual-inertial fusion,” *Journal of Field Robotics*, vol. 35, no. 1, pp. 23–51, 2018.
- [79] R. Mur-Artal and J. D. Tardos, “Visual-inertial monocular slam with map reuse,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, Apr. 2017.
- [80] E. D. Nerurkar, K. J. Wu, and S. I. Roumeliotis, “C-KLAM: Constrained keyframe-based localization and mapping,” *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 3638–3643, 2014.
- [81] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, “Keyframe-Based Visual-Inertial SLAM Using Nonlinear Optimization,” *Proc. of Robot.: Sci. and Syst.*, p. 0, 2013.
- [82] H. Kretzschmar and C. L. Stachniss, “Information-theoretic compression of pose graphs for laser-based slam,” *International Journal of Robotics Research*, vol. 31, no. 11, pp. 1219–1230, 2012.
- [83] N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice, “Generic node removal for factor-graph slam,” *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1371–1385, 2014.
- [84] V. Paksoy, R. Türkmen, and F. Zhang, “Inequalities of generalized matrix functions via tensor products,” *The electronic journal of linear algebra ELA*, vol. 27, p. 332, 03 2014.
- [85] T. Pire, “Stereo parallel tracking and mapping,” Ph.D. dissertation, University of Buenos Aires, 2016.
- [86] S. Agarwal, K. Mierle, and Others, “Ceres solver,” <http://ceres-solver.org>.
- [87] D. Gálvez-López and J. D. Tardos, “Bags of binary words for fast place recognition in image sequences,” *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.
- [88] R. Mur-Artal and J. D. Tardós, “Fast relocalisation and loop closing in keyframe-based slam,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 846–853.

-
- [89] G. Dudek and M. Jenkin, *Computational principles of mobile robotics*. Cambridge university press, 2010.
- [90] L. Palmer and M. Palankar, “Blind hexapod walking over uneven terrain using only local feedback,” in *2011 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2011, pp. 1603–1608.
- [91] A. Winkler, I. Havoutis, S. Bazeille, J. Ortiz, M. Focchi, R. Dillmann, D. Caldwell, and C. Semini, “Path planning with force-based foothold adaptation and virtual model control for torque controlled quadruped robots,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6476–6482.
- [92] T. Krajník, M. Nitsche, J. Faigl, P. Vaněk, M. Saska, L. Přeučil, T. Duckett, and M. Mejail, “A practical multirobot localization system,” *Journal of Intelligent & Robotic Systems*, 2014. [Online]. Available: <http://dx.doi.org/10.1007/s10846-014-0041-x>
- [93] E. Olson, “AprilTag: A robust and flexible visual fiducial system,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.
- [94] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, “An evaluation of the RGB-D SLAM system,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 1691–1696.
- [95] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in *ICRA Workshop on Open Source Software*, 2009.
- [96] J. Civera, A. J. Davison, and J. M. Montiel, “Inverse depth parametrization for monocular slam,” *IEEE transactions on robotics*, vol. 24, no. 5, pp. 932–945, 2008.
- [97] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, “Rslam: A system for large-scale mapping in constant-time using stereo,” *International Journal of Computer Vision*, vol. 94, no. 2, pp. 198–214, Sep. 2011.
- [98] V. Ila, L. Polok, M. Solony, and P. Svoboda, “Slam++1-a highly efficient and temporally scalable incremental slam framework,” *International Journal of Robotics Research*, vol. 36, no. 2, pp. 210–230, 2017.