UNIVERSIDAD
DE MÁLAGA

Doctoral Dissertation

Contributions to metric-topological localization and mapping in mobile robotics

Eduardo Fernández-Moral

2014

Tesis doctoral en Ingeniería Mecatrónica
Dpto. de Ingeniería de Sistemas y Automática
Universidad de Málaga

UNIVERSIDAD DE MÁLAGA
DEPARTAMENTO DE
INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

El Dr. D. Javier González Jiménez y el Dr. D. Vicente M. Arévalo Espejo, directores de la tesis titulada "Contributions to metric-topological localization and mapping in mobile robotics" realizada por D. Eduardo Fernández-Moral, certifican su idoneidad para la obtención del título de Doctor en Ingeniería Mecatrónica.

Málaga, 10 de Septiembre de 2014

Dr. D. Javier González Jiménez

Dr. D. Vicente M. Arévalo Espejo

Dept. of System Engineering and Automation
University of Málaga
Studies in Mechatronics

UNIVERSIDAD
DE MÁLAGA

Contributions to metric-topological localization and mapping in mobile robotics

AUTHOR: Eduardo Fernández Moral

SUPERVISORS: Javier González Jiménez
Vicente M. Arévalo Espejo

A mi tío Hilario, por mostrarme el camino...

Acknowledgements

First, I would like to express my immense gratitude to my supervisors Prof. Dr. Javier González Jiménez and Dr. Vicente Arévalo for guiding me through the hard but exciting adventure of working for a PhD. They are responsible of the invaluable technical and theoretical knowledge that I have acquired, which gives me professional independence. The uncountable hours we have spent together discussing about different research problems and their results have today a reward in this thesis. Also, they were always comprehending at the hard moments of this PhD, offering their best. I also have to thank Guillaume Charpiat and Alexander Davies for the time they spent reviewing this thesis and for the useful comments they have provided me. I am also grateful to the committee members for having accepted to be members of this thesis' tribunal.

These four years of research probably would not have been successful without the great people in our lab. Thanks to Raul Ruiz, Javier G. Monroy, Ana Gago, Francisco Moreno, Mariano Jaimez, Francisco Meléndez, Carlos Sánchez, Manuel López, Rubén Gómez and Jesús Briales for being good colleagues and friends. Thanks to them for their support and for creating an atmosphere of joy where it was a pleasure to work. I also have to thank the senior researchers at the MAPIR group, Cipriano Galindo and Juan Antonio Fernández, and to our former colleague Jose Luis Blanco for sharing always their knowledge and expertise when I had needed it.

During 2012 I had the opportunity to work as a research visitor at the Vision Information Laboratory at the University of Bristol. I acknowledge Dr. Walterio Mayol Cuevas and his team for giving me this opportunity. I do not forget José Martínez Carranza for the fruitful discussions during this period. Also, my deep gratitude goes to Dr. Patrick Rives who received me at the Lagadic Team, at INRIA Sophia-Antipolis for a research visit during 2013. I need to thank as well the rest of the staff and PhD students in his team to make me feel like at home during this period.

Finally, I feel extremely grateful to my family, which has always supported me along this long period at university and during my PhD. And I will not forget to mention my good friends Manuel J. García and Manuel J. Nájera because despite my research work keeps me far from them, I know that they will be there when I need them.

Summary

This thesis addresses the problem of localization and mapping in mobile robotics. The ability of a robot to build a map of an unknown environment from sensory information is required to perform self-localization and autonomous navigation, as a necessary condition to carry out more complex tasks. This problem has been widely investigated in the last decades, but the solutions presented have still important limitations, mainly to cope with large scale and dynamic environments, and to work in a wider range of conditions and scenarios. In this context, this thesis takes a step forward towards highly efficient localization and mapping.

A first contribution of this work is a new mapping strategy that presents two key features: the lightweight representation of world metric information, and the organization of this metric map into a topological structure that allows efficient localization and map optimization. Regarding the first issue, a map is proposed based on planar patches which are extracted from range or RGB-D images. This plane-based map (PbMap) is particularly well suited for indoor scenarios, and has the advantage of being a very compact and still a descriptive representation which is useful to perform real-time place recognition and loop closure. These operations are based on matching planar features taking into account their geometric relationships. On the other hand, the abstraction of metric information is necessary to deal with large scale SLAM and with navigation in complex environments. For that, we propose to structure the map in a metric-topological structure which is dynamically organized upon the sensor observations.

Also, a simultaneous localization and mapping (SLAM) system employing an omnidirectional RGB-D device which combines several structured-light sensors (Asus Xtion Pro Live) is presented. This device allows the quick construction of rich models of the environment at a relative low cost in comparison with previous alternatives. Our SLAM approach is based on a hierarchical structure of keyframes with a low level layer of metric information and several topological layers intended for large scale SLAM and navigation. This SLAM solution, which makes use of the metric-topological representation mentioned above, works at video frame rate obtaining highly consistent maps. Future research is expected on metric-topological-semantic mapping from the new sensor and the SLAM system presented here.

Finally, an extrinsic calibration technique is proposed to obtain the relative poses of a combination of 3D range sensors, like those employed in the omnidirectional RGB-D device mentioned above. The calibration is computed from the observation of planar surfaces of a structured environment in a fast, easy and robust way, presenting qualitative and quantitative advantages with respect to previous approaches. This technique is extended to calibrate any combination of range sensors, including 2D and 3D range sensors, in any configuration. The calibration of such sets of sensors is interesting not only for mobile robots, but also for autonomous cars.

Contents

Acknowledgements **i**

Summary **iii**

Contents **v**

Resumen de la Tesis Doctoral **1**

1 Introduction **9**

1.1 Motivation 9
1.2 Scope of this thesis 11
1.3 Contributions 12
1.4 Framework of this thesis 14
1.5 Structure of this thesis 15

2 Calibration of sensor rigs **17**

2.1 Introduction 18
2.2 Intrinsic calibration 19
2.2.1 Calibration of a regular camera (RGB) 19
2.2.2 Calibration of depth cameras 21
2.3 Extrinsic calibration of range sensors 22
2.3.1 Calibration of several 2D laser rangefinders 23
2.3.2 Calibration of a set of 3D range cameras 44
2.3.3 Calibrating a 3D range camera and a 2D laser scanner 56
2.4 Conclusions 66

3 Plane-based maps for fast localisation and place recognition **67**

3.1 Introduction 68
3.1.1 Related works 70
3.1.2 Contribution 73
3.2 Plane segmentation 74
3.3 Formal definition of a PbMap 75

3.4 Compact colour descriptor 76
3.4.1 Colour information for patch matching 77
3.4.2 Selection of the colour space 79
3.4.3 Computing the dominant colour 80
3.4.4 Choosing a robust colour descriptor 82
3.4.5 Comparison with the hue histogram 83
3.5 PbMap construction 84
3.6 Place recognition and localization in PbMaps 85
3.6.1 Choosing the scope of search 85
3.6.2 Graph matching 86
3.6.3 Localization and rigid consistency 91
3.7 Experimental validation 92
3.7.1 Recognition from range images 93
3.7.2 Recognition from RGB-D images 94
3.8 Discussion 96

4 Hybrid metric-topological mapping 101
4.1 Introduction 102
4.1.1 Related works 103
4.1.2 Contribution 104
4.2 Hybrid metric-topological mapping approach 105
4.2.1 SLAM Improvements through hybrid mapping 105
4.2.2 Map partitioning method 106
4.3 Metric-topological monocular SLAM 108
4.3.1 Keyframes selection in large environments 109
4.3.2 Combination of map partitioning and PTAM 112
4.4 Experiments 113
4.5 Discussion 115

5 A SLAM system for omnidirectional RGB-D sensors 117
5.1 Introduction 118
5.1.1 Related works 119
5.1.2 Contribution 120
5.2 Omnidirectional RGB-D device 121
5.2.1 Sensor set-up 121
5.2.2 Calibration 123
5.2.3 Spherical representation 123
5.3 SLAM approach 125
5.3.1 Localization 125
5.3.2 Mapping 128
5.3.3 Loop closure 130
5.4 Experimental validation 131
5.4.1 Fast scene registration 131
5.4.2 Keyframe-based SLAM 133

5.5 Discussion 135

6 Conclusions 137

Appendices 139

A Maximum Likelihood Estimation and Least Squares 141

B Lie algebra and Lie groups 145

C Propagation of uncertainty 147

D Fisher Information and Cramér-Rao Bound 149

Bibliography 151

Resumen de la Tesis Doctoral

Sumario

La presente tesis doctoral aborda el problema de localización y cartografía (o mapeo) en robótica móvil. La capacidad de un robot móvil para crear un mapa de su entorno a partir de la información obtenida por sus sensores es necesaria para que dicho robot pueda localizarse y para que éste pueda navegar de forma autónoma. Este problema ha sido ampliamente estudiado en las últimas décadas, sin embargo, las soluciones obtenidas presentan aún importantes limitaciones. Estas limitaciones afectan principalmente a la operación en entornos a gran escala y en la adaptación a entornos dinámicos. En este contexto, esta tesis representa otro paso más en el camino hacia soluciones de localización y mapeo eficientes.

Una primera contribución de este trabajo es una nueva estrategia de cartografía que combina dos características principales: una representación compacta de la información métrica, y la organización de esta información métrica en una estructura topológica que mejora la eficiencia en la localización y la optimización del mapa. Para ello se propone un mapa basado en segmentos planos que son extraídos de las imágenes de rango o RGB-D. Este mapa basado en planos (PbMap) es especialmente adecuado para escenarios de interior, y tiene la ventaja de ser altamente descriptivo a pesar de su compacidad, lo cuál permite reconocer escenarios en tiempo real y cerrar bucles, siendo éstas tareas clave para la localización y mapeo simultáneos (SLAM). Ambas operaciones se basan en el emparejamiento de segmentos planos teniendo en cuenta sus relaciones geométricas. Por otro lado, la abstracción de la información métrica es necesaria para abordar el problema de SLAM en gran escala y para la navegación en entornos complejos. En este sentido, esta tesis propone organizar el mapa en tiempo real en una estructura métrica-topológica de acuerdo a la co-visibilidad de las observaciones.

Esta tesis también presenta un sistema de localización y cartografía simultánea (SLAM) empleando un nuevo sensor omnidireccional RGB-D que combina varios sensores Asus Xtion Pro Live. Este dispositivo permite construir rápidamente modelos densos del entorno basados en nubes de puntos a un bajo coste con respecto a

otras alternativas previas. Nuestro enfoque en SLAM se basa en la gestión de una estructura jerárquica de *keyframes* consistente en una capa de bajo nivel con información métrica y varias capas superiores con información topológica que son útiles para SLAM en gran escala y para navegación. Este sistema de SLAM funciona a una frecuencia de 30 Hz y permite la obtención de mapas de alta consistencia.

También se propone una técnica de calibración extrínseca para obtener las poses relativas de una combinación de sensores de rango 3D, como aquellos empleados en el dispositivo RGB-D omnidireccional mencionado anteriormente. La calibración se obtiene a partir de la observación de superficies planas en un entorno estructurado de una manera rápida, fácil y robusta. Esta nueva técnica presenta ventajas cualitativas y cuantitativas con respecto a los enfoques anteriores. Esta solución es ampliada para calibrar cualquier combinación de sensores de rango en cualquier configuración, incluyendo sensores 2D y 3D. La calibración de tales conjuntos de sensores es interesante no sólo en robótica móvil, sino también en el campo de vehículos autónomos.

Introducción

En los últimos años del siglo XX y principios del XXI se había generalizado en el mundo desarrollado la impresión de que en la época actual viviríamos rodeados de robots inteligentes que harían nuestras vidas *más fáciles*, o deberíamos decir, que nos ahorrarían tediosas tareas rutinarias. Las películas de ciencia ficción han contribuido a crear dicha imagen de nuestro futuro, en el que convivimos con complejos robots móviles. Por otro lado, los avances en microelectrónica, el aumento del rendimiento de computadores, y la aparición de nuevos materiales y sensores, que hoy tienen sus resultados en los actuales teléfonos móviles (o *smartphones*) por ejemplo, también han ayudado a pensar que la robótica móvil aparecería pronto en nuestras vidas. Sin embargo la realidad sigue siendo muy diferente de aquella imagen futurista, la cuál necesitará probablemente un largo tiempo para hacerse realidad.

Una de las principales razones para no tener robots móviles entre nosotros es la dificultad para procesar e interpretar información del entorno del robot. Esto es clave para la robótica móvil ya que un robot debe *entender* su entorno para poder interactuar con él. En este contexto, la capacidad de un robot móvil para crear un mapa de su entorno al mismo tiempo que se localiza en dicho mapa es crucial. Este problema, conocido como localización y cartografía simultáneas (SLAM), ha recibido una gran atención en las últimas décadas, dando como resultado una amplia literatura sobre este tema que abarca diferentes condiciones de trabajo y diferentes tipos de sensores. Sin embargo, a pesar del gran esfuerzo dedicado a solventar este problema, las soluciones presentadas tienen todavía importantes limitaciones que impiden la creación de robots fiables que puedan ejecutar tareas útiles en condiciones generales.

La investigación en SLAM se ha centrado principalmente en el uso de dos tipos de sensores exteroceptivos: cámaras y sensores de rango. Las cámaras presentan importantes ventajas como su bajo coste, su compacidad y su bajo consumo, lo que las hace adecuadas para diversas aplicaciones en robótica móvil además de SLAM, como el reconocimiento de objetos. Además, las cámaras proporcionan información similar a la captada por nuestros ojos, siendo por tanto una herramienta intuitiva de percepción. Sin embargo, la información real proporcionada por una cámara es una matriz rectangular donde cada celda (pixel) captura la luminosidad procedente de una dirección del espacio. Por lo tanto, el primer problema es representar esta información de una manera compacta y estructurada que pueda ser interpretada fácilmente. La mayoría de las soluciones encontradas en la literatura abordan este problema mediante la extracción de diferentes tipos de características de bajo nivel que pueden ser identificadas desde diferentes puntos de vista a lo largo de la trayectoria del robot. Otros enfoques recientes tratan de identificar objetos significativos para ser utilizados como referencias en el entorno. En cualquier caso, el problema de SLAM visual implica algunas limitaciones intrínsecas para operar en escenarios con poca textura, con repetición de texturas (*visual aliasing*), o donde existen reflejos especulares.

Con respecto a los sensores de rango, la percepción de profundidad permite a un robot crear representaciones del espacio y evitar colisiones, creando mapas útiles para la auto-navegación y para llevar a cabo el reconocimiento de objetos y lugares. Existen distintas maneras de obtener información de profundidad del entorno, en el que podemos diferenciar entre los métodos activos o pasivos. Las estrategias activas proyectan y capturan luz de la escena para inferir la profundidad, mientras que las estrategias pasivas sólo capturan luz. Ejemplos de este último son los sistemas de visión estéreo o multi-cámaras; mientras que algunos ejemplos de visión activa son LIDAR, cámaras de tiempo de vuelo y sensores basados en luz estructurada como Kinect. Diferentes soluciones se han presentado para SLAM con un robot que se mueve sobre un mismo plano en un entorno estático utilizando un sensor 2D. El uso de sensores 3D también ha sido introducido para operar en condiciones más complejas. Para este último, algunas desventajas comunes son el alto precio de los sensores, las bajas frecuencias de observación o la escalabilidad de las soluciones de SLAM.

El problema de SLAM también se ha tratado combinando información visual y de rango, especialmente después de la aparición de cámaras RGB-D de bajo coste como Asus Xtion Pro Live o Microsoft Kinect. La fusión de la información de profundidad e intensidad mejora la capacidad de SLAM proporcionando robustez a situaciones donde la intensidad o la profundidad por sí solas tienen un bajo rendimiento. Dichos sensores se han empleado por ejemplo para odometría mediante el registro denso de las imágenes RGB-D. Cuando estos sensores se utilizan en SLAM, uno de los principales problemas es cómo representar y almacenar el gran flujo de datos que éstos proporcionan. Differentes estrategias de cartografía basadas en *keyframes* han proporcionado buenos resultados en pequeños entornos, pero aún existen problemas de escalabilidad y por lo tanto, representaciones del entorno más compactas son deseables para realizar otras tareas junto con SLAM.

Con respecto a la robótica móvil, los desafíos actuales en localización y mapeo están relacionados principalmente con limitaciones en el tamaño del entorno de trabajo y con la fiabilidad de funcionamiento a lo largo del tiempo. Otro problema clave es la integración de información simbólica para aumentar la robustez y el rendimiento a la vez que se proporciona información útil para otros tareas. Pero como se puede intuir, estos desafíos requieren avances incrementales en las soluciones actuales hasta llegar al objetivo de localización y mapeo robusto en condiciones más generales.

Ámbito de la tesis

La investigación del problema de localización y mapeo simultáneo ha recibido una amplia atención en los últimos años y se han presentado diferentes soluciones al problema. En este contexto, la contribución de esta tesis debe ser considerada como un paso más en un largo camino hacia la obtención de soluciones más generales, robustas y eficientes para SLAM, que doten a los robots de autonomía real en una variedad de escenarios. El problema de SLAM implica diferentes subproblemas, desde la calibración de los sensores del robot a la representación de la información en el mapa junto con la localización y re-localización (cierre de bucle) eficientes. Estos problemas se abordan en los capítulos siguientes mediante el uso de diferentes sensores visuales y de rango.

Concretamente, esta tesis presenta una nueva metodología para la calibración de conjuntos de sensores de rango que está basada en la observación de superficies planas. Dicha metodología es utilizada para calibrar un nuevo sensor que consta de varias cámaras de RGB-D, permitiendo también calibrar otras combinaciones de sensores de rango con escáneres láser 2D y sensores de rango 3D con los que están equipados muchos robots móviles, incluyendo los robots empleados en esta tesis. La observabilidad de los diferentes problemas (dependiendo del tipo de sensores) es analizada, y se definen las condiciones para resolver la calibración extrínseca a partir de un conjunto mínimo de observaciones. En todos los casos, la solución propuesta permite calibrar los sensores fácilmente y de forma robusta después de unos segundos observando una escena estructurada.

Uno de los principales retos en SLAM es cómo extraer las características más útiles de la escena para mantener una representación compacta de esta, descartando al mismo tiempo información redundante. Esto es necesario para operar eficientemente en tiempo real, tal como se requiere para muchas aplicaciones de robótica móvil. Para ello presentamos un mapa métrico basado en la extracción de superficies planas de la escena, que almacena un conjunto de características geométricas y radiométricas de manera compacta. Dicha representación ha demostrado ser útil para el registro robusto de imágenes, para odometría usando cámaras de rango y para el reconocimiento automático de lugares.

El uso de superficies planas para la localización y mapeo presenta ventajas en cuanto a la reducción de memoria y procesamiento. Por el contrario, existen limitaciones de inobservabilidad cuando no hay suficientes planos visibles, siendo la localización ambigua. Esta restricción puede ser resuelta en general aumentando el campo de visión de los sensores utilizados. En esta línea, esta tesis presenta un nuevo dispositivo para capturar imágenes omnidireccionales RGB-D a una frecuencia de 30Hz. Las imágenes capturadas por este dispositivo son adecuadas para su representación en coordenadas esféricas, lo cual ofrece una serie de ventajas como un mejor condicionamiento de la localización, el desacople natural entre la rotación y la traslación, la creación de mapas compactos basados en *keyframes*, o su adecuación para clasificación topológica de imágenes.

La organización del mapa es una cuestión clave para el funcionamiento de SLAM en gran escala. La literatura sobre este tema es amplia, y se pueden encontrar diferentes estrategias que proponen estructuras topológicas, métrico-topológicas o jerárquicas para los mapas. La necesidad de este tipo de estructuras se justifica porque un sistema SLAM para gran escala debe abstraerse de la información que no es significativa (por ejemplo, teniendo solo en cuenta información métrica relativa a la localización actual del robot). En esta tesis, se presenta una estrategia para organizar dinámicamente la información métrico-topológica en tiempo real que agrupa las observaciones que están más interrelacionadas, formando lugares topológicos. Esta estructura mejora la eficiencia y permite la escalabilidad en SLAM.

Por último, esta tesis presenta un nuevo enfoque en SLAM con el uso de imágenes omnidireccionales RGB-D que combina los avances descritos arriba en cuanto a calibración, localización y mapeo. Re-localización y cierre de bucle son dos problemas inherentes en SLAM que se tratan aquí. El primero se refiere a la capacidad para estimar la ubicación del robot cuando se ha perdido la localización (un problema similar es conocido como *robot awakening*), mientras que el segundo implica que el robot pueda reconocer una ubicación previamente visitada a través de una trayectoria diferente. La detección del cierre de bucle permite reducir la incertidumbre del robot y mejorar la coherencia global del mapa. Ambos problemas se abordan en esta tesis mediante el uso de una representación de la escena basada en planos.

Conclusiones

Las contribuciones más relevantes de esta tesis son:

- Una nueva metodología para calibrar diferentes tipos de sensores de rango basada en la observación de superficies planas. Esta metodología permite calibrar los parámetros extrínsecos de los sensores de forma fácil y robusta en unos pocos segundos [Fernández-Moral *et al.*, 2014b], [Fernández-Moral *et al.*, 2015b].

- Una representación del entorno altamente compacta basada en superficies planas que sintetiza información geométrica y radiométrica (PbMap). Esta representación es útil para el modelado de entornos estructurados, para la localización del robot y para la detección del cierre de bucle [Fernández-Moral *et al.*, 2013b], [Fernández-Moral *et al.*, 2014a].

- Una técnica de registro para PbMaps basado en el emparejamiento de conjuntos de planos vecinos mediante un árbol de interpretación. Esta técnica no restringe el emparejamiento a una sola imagen, sino que cualquier conjunto local de los planos es válido para ser emparejados lo que permite usar la información de varias observaciones [Fernández-Moral *et al.*, 2013b].

- Una estrategia de mapeo métrico-topológica, basada en corte normalizado de grafos, que re-organiza dinámicamente el mapa en diferentes regiones topológicas mientras este se actualiza simultáneamente. Esta estrategia de mapeo permite la operación de SLAM en gran escala y ofrece ventajas para la navegación y la planificación de tareas [Fernández-Moral *et al.*, 2013a], [Fernández-Moral *et al.*, 2015a].

- El desarrollo de un nuevo sensor para la adquisición de imágenes omnidireccionales RGB-D a 30Hz consistente en un conjunto de 8 sensores Asus Xtion Pro Live montados en una configuración radial [Fernández-Moral *et al.*, 2014b], junto con un nuevo sistema de SLAM basado en un mapa métrico-topológico de *keyframes* [Gokhool *et al.*, 2014].

Todas las publicaciones derivadas de esta tesis están disponibles en: `http://mapir.isa.uma.es`

Marco de esta tesis

Esta tesis es el resultado de cuatro años de actividad investigadora de su autor como miembro del grupo de investigación MAPIR[1], dentro del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga. Esta investigación ha sido financiada por el Gobierno español a través del “Fondo Regional de Desarrollo Europeo FEDER” dentro de los contratos DPI2008-03527 y DPI2011-25483, en los que se enmarcan los proyectos “Construcción de mapas topológicos métrica-visuales para robótica móvil” y “TAROTH: Nuevos avances hacia un robot en el hogar", respectivamente. El primer proyecto enfoca la creación de una representación del entorno para una variedad de sensores visuales, y comprende los dos primeros años de investigación de esta tesis. El segundo comprende el resto de esta tesis, y abarca la calibración de conjuntos de sensores y la explotación de los mapas previos para aplicaciones de localización y SLAM.

Durante el doctorado el autor completó el programa doctoral titulado “Ingeniería Mecatrónica” coordinado por el Departamento de Ingeniería de Sistemas y Automática

[1] http://mapir.isa.uma.es

de la Universidad de Málaga. Este programa de doctorado le proporciona al autor una visión general del campo multidisciplinar de la mecatrónica que combina mecánica, eléctrica, control e ingeniería informática, y lo más importante un conocimiento profundo acerca de la robótica móvil, algo que ha resultado fundamental a lo largo de estos años de investigación. Además, el autor ha completado su formación académica con su participación en el curso de Visión por Computador (BMVA 2010) de la Universidad de Kingston, Londres.

El autor desarrolló parte de su investigación en colaboración con dos grupos de investigación internacionales. En 2012, estuvo 4 meses con el grupo Visual Information Laboratory en la Universidad de Bristol (Reino Unido), bajo la supervisión de Dr. Walterio Mayol-Cuevas. Durante este período, su investigación se centró en la explotación de estructuras planas para la construcción de mapas y SLAM. En 2013, estuvo 9 meses con el equipo Lagadic en INRIA Sophia-Antipolis (Francia), bajo la supervisión de Dr. Patrick Rives. Durante este tiempo, el autor trabajó en el desarrollo de un dispositivo RGB-D omnidireccional concebido para la construcción de mapas y la navegación de robots.

Por último, es necesario mencionar que el marco científico de esta tesis es un área de investigación muy competitiva, que es impulsada por una creciente industria en visión por computador y robótica. En la opinión del autor, las continuas contribuciones en estas dos áreas que están altamente interrelacionadas permitirán la progresiva integración de robots móviles en nuestra sociedad.

Estructura de la tesis

Con el objetivo de obtener la mención de Doctorado Internacional por la universidad de Málaga, el desarrollo completo de esta tesis está escrito en español e inglés. Así, el texto está dividido en dos partes. La primera parte, escrita en español, describe de forma resumida el contenido del trabajo, mientras que la segunda parte, redactada íntegramente en inglés, presenta una descripción completa del mismo. Esta segunda parte se compone de los siguientes capítulos:

El **capítulo 2** introduce los conceptos básicos relativos a la calibración de diferentes sensores, y aporta una nueva metodología para calibrar conjuntos de sensores de rango. La estrategia presentada no requiere ningún patrón específico ya que se basa en la detección de superficies planas del entorno. Esta técnica de calibración es rápida, fácil de usar y robusta, presentando importantes ventajas con respecto a alternativas previas.

El **capítulo 3** propone una nueva representación de la escena basada en superficies planas que son segmentadas de imágenes de rango. Este mapa basado en planos (PbMap) es descrito por un grafo que contiene una serie de características geométricas y radiométricas. También se propone un descriptor compacto de color basado en el color dominante del plano que contribuye a la eficiencia y robustez en el emparejamiento de planos. Una técnica de reconocimiento de escenas es propuesta basada en el registro de tales planos. Los resultados cuantitativos y cualitativos aportados en el ámbito de reconocimiento de lugar confirman las ventajas de esta representación.

El **capítulo 4** aborda el problema de la cartografía métrico-topológica. La estructura topológica se representa mediante un grafo no dirigido donde los nodos contienen información métrica y los arcos definen la conectividad entre regiones locales. Esta estructura tiene ventajas para el manejo eficiente del mapa, ya que sólo la información métrica local es utilizada para la localización y mapeo. Este capitulo propone una estrategia dinámica para gestionar el mapa, donde las diferentes regiones locales son agrupadas en función de la interconexión de las diferentes observaciones. Esta técnica es evaluada en el marco de SLAM monocular (usando PTAM [Klein and Murray, 2007]) y de SLAM omnidireccional RGB-D (capítulo 5).

El **capítulo 5** presenta un nuevo sensor para capturar imágenes omnidireccionales RGB-D a 30 Hz, junto con una solución SLAM para este tipo de sensor. El enfoque SLAM se basa en una estructura métrico-topológica de *keyframes* que se organizan en una red jerárquica de mapas locales. Los *keyframes* se describen a través de un PbMap que es utilizado para la localización y para el cierre de bucle eficientes. La localización obtenida del registro de PbMaps es refinada por una técnica de registro denso. La consistencia del mapa se mejora mediante la optimización de mapa global teniendo en cuenta todas las conexiones de los *keyframes*.

El **capítulo 6** concluye la tesis, proporcionando un resumen de la investigación presentada y expone el panorama de los retos futuros para la localización autónoma y la cartografía. En este contexto, en el futuro se espera la continuación de la investigación llevada a cabo en esta tesis en diferentes aspectos. Por un lado, el tratamiento probabilístico de la representación mediante PbMap supondría un avance en precisión, que además permitiría fusionar de forma coherente los planos observados por sensores diferentes. Por otro lado, la incorporación de información semántica al PbMap y a los diferentes niveles topológicos en los que se estructura el mapa son otra línea de investigación que previsiblemente ganará popularidad en los años venideros.

Chapter 1
Introduction

1.1 Motivation

Mobile robots are far from the state of development imagined a few years ago in our society. There has existed the general impression in the late years of the 20th century and beginning of the 21st, that intelligent robots would be spread in our society making our lives *easier*, or should we say, releasing us from tiresome routines. But the reality is still quite different. Science fiction films have contributed to build such an image of our future, where we live side by side with intelligent mobile robots. Also, the rapid technological development in the miniaturization of electronics, the increase of processors' computing performance, and the appearance of new materials and sensors, which today have their results in compact smartphones for instance, suggested that mobile robotics would come along undoubtedly. But unfortunately, such a futuristic image will need some more time to come true.

One of the main reasons for not having mobile robots around us nowadays is the difficulty to process and interpret exteroceptive information from the world. This is key for mobile robotics since a robot must *understand* its environment before it can interact with it. In this context, the ability of a mobile robot to create a map of its environment at the same time that it performs self-localization within such a map becomes crucial. This problem, known as simultaneous localization and mapping (SLAM), has received great attention during the last decades, and there exists a vast literature on the subject for a variety of working conditions using different sensors. However, despite the large effort dedicated to this topic, the solutions presented have still important limitations that prevent robots to work reliably in uncontrolled conditions.

Research in SLAM has mainly focused on two kinds of data: photometric information from regular cameras or multi-camera systems, and depth information from range sensors. The use of cameras have some nice advantages as they are inexpensive, compact and consume low power, what makes them suitable for other applications in mobile robotics besides SLAM, e.g. object recognition. Also, cameras provide information similar to that captured by our eyes, being thus an intuitive way to perform robot perception. However, the actual information provided by a camera is a rectangular matrix where each cell (pixel) captures the luminosity coming from one direction of the space, hence, the first problem regarding localization is to represent this information in a more compact and structured way which provides invariance to different viewpoints and illumination conditions. Most solutions found in the literature address this problem by extracting different kinds of low level features which can be matched along the robot trajectory. While some modern approaches also try to identify meaningful objects to be used as such features. Several limitations are intrinsically related to visual SLAM, like the operation in low texture scenarios, visual aliasing or specular reflections.

On the other hand, depth perception allows a robot to create representations of the free space and to avoid collisions, thus creating useful maps for self-navigation, and to perform recognition through shape description. There exist different ways of acquiring depth information from the environment, where we can differentiate between active or passive approaches. Active strategies project and capture light from the scene to perform depth inference, while passive strategies only capture light. Examples for the latter are stereo vision or multi-camera systems; while active vision examples are LIDAR, time-of-flight cameras (ToF camera) and structured light sensors. Many SLAM approaches have been presented for planar movement of a robot in a static environment using a 2D range sensor. The use of 3D range devices has also been presented to expand the utility of 2D approaches in more complex conditions. For the latter, some common disadvantages are the high price of the sensors, the low frame rate, the difficulty to obtain scalable solutions and the unobservability of localization depending on the sensor/environment which may affect severely the robustness of the solution (e.g. a sensor which only observes the floor).

Combined visual and range SLAM approaches have also been exploited, especially after the release of low cost RGB-D cameras like Asus Xtion or Microsoft Kinect. The fusion of depth and intensity information enhances the capability of SLAM by making it more robust to situations where only intensity or only depth approaches have a low performance. This data also permits to create dense coloured 3D point maps of the environment that provide nice visualizations. Such sensors have been employed for example for odometry by applying dense alignment of the RGB-D images. When such sensors are to be used for SLAM, the main problem is how to represent and store the big data streaming they provide. Mapping strategies based on keyframes, inspired by those created for visual SLAM, have provided nice results with hand-held sensors, but there are still scalability issues, and more compact representations of the environment are desired so that other robotic tasks can be performed on the same computer.

Regarding its application to real mobile robots, the current challenges in localization and mapping are mainly related to size and time scalability (lifelong mapping), and the integration of symbolic information to increase the robustness and performance at the same time that provides useful information for other tasks. But as we can guess, these challenges require little advances to progress towards the goal of reliable localization and mapping.

1.2 Scope of this thesis

The problem of simultaneous localization and mapping has centred the attention of an extensive research over the last years, and many approaches to the problem have been presented. In such a context, the contribution of this thesis must be regarded as another step in a long way towards finding a more general, highly robust and efficient approach for SLAM, which gives a robot real autonomy in a variety of scenarios. The problem of SLAM can be divided in different subproblems, from sensor calibration to efficient map representation and localization, including loop closure. These problems are addressed in the next chapters using different visual and range sensors.

Concretely, this thesis presents a new methodology for calibrating sets of range sensors based on the observation of planar surfaces at different orientations. Such methodology is used here to calibrate a new sensor introduced in chapter 5 which is composed of several depth cameras, and is also extended to calibrate other combinations of range sensors like sets of 2D laser scanners, and combinations between 2D and 3D range sensors which are present in the robots employed in the thesis and in many other robotic set-ups in general for navigation and SLAM. The observability of the different problems (depending on the type of sensors) is analysed, providing the conditions to solve the extrinsic calibration from a minimum set of observations. In all the cases, the proposed solution permits to compute the calibration easily and robustly after a few seconds taking measurements.

One of the main challenges for SLAM, and still an open problem, is how to extract the most useful cues from the scene which permits to keep a compact representation of the scene, while discarding redundant information. This compact representation is required to operate efficiently in real-time, as required for many mobile robotics applications. For such a problem, we present a (metric) mapping approach based on the extraction of planar surfaces from the scene, which stores a set of geometric and radiometric cues in a compact fashion. Such representation has demonstrated to be useful for robust image registration, odometry and place recognition. This mapping approach is presented in chapter 3.

Using only planar surfaces for localization and mapping presents advantages for low memory storage and fast processing. However, unobservability limitations arise when there are not enough planes in view, so that localization becomes ambiguous. This restriction can be alleviated, while maintaining our compact representation, by increasing the field of view of our sensing technology. In this line, we present a new device to capture omnidirectional RGB-D images online. The images captured by this device are suitable for spherical representation, which offers a number of advantages for different applications, as well conditioned localization, compact keyframe mapping, or topological image classification.

The organization of the map becomes a key question for scalable operation in SLAM. A rich literature can be found about this subject, where large scale SLAM is addressed with topological, hierarchical or hybrid (metric-topological) maps. The need of such structures is justified because an effective large scale SLAM system must abstract itself from information which is less meaningful (so that only the map information related to the current robot localization is taken into account). In this thesis, we present an online metric-topological map arrangement where scarcely interrelated parts of the map are separated into different groups, forming topological places. This map structure boosts efficiency and allows for scalable SLAM operation with sublinear loop closure. Chapter 4 is dedicated to such metric-topological mapping.

Finally, a new SLAM approach is presented using omnidirectional RGB-D images, which combines the developments described in previous chapters. Relocalization and loop closure are inherent problems of SLAM which are treated here. The first refers to the ability to retrieve the location of the robot when it has got lost (i.e. awakening problem), while the second implies that the robot can detect a previous location when it is revisited through a different trajectory. Loop closure detection allows the reduction of uncertainty in the robot location and to improve the global consistency of the map. Both problems are addressed in this thesis by using the proposed plane-based representation of the scene.

1.3 Contributions

The main contributions of this thesis are:

- A new methodology to calibrate different kinds of range sensors based on planar surface observations. This methodology permits to calibrate the extrinsic parameters of the sensors easily and robustly in a few seconds [Fernández-Moral *et al.*, 2014b] and [Fernández-Moral *et al.*, 2015b].

- A highly compact map representation based on planar patches (PbMap), which synthesizes geometric and radiometric information. This representation is useful for Manhattan-like modelling, for efficient robot localization and for loop closure detection among others [Fernández-Moral *et al.*, 2013b; Fernández-Moral *et al.*, 2014a].

- A registration technique for PbMaps based on graph matching of local contexts of planes. An interpretation tree is used to match efficiently the planes of both graphs. This technique does not restrict the matching to one image, instead, any local set of planes which are distinctive are valid to be matched [Fernández-Moral *et al.*, 2013b].

- A metric-topological mapping framework, based on approximate minimum normalized cut, which dynamically re-organizes the metric map into different topological regions simultaneously while building the map. This mapping strategy makes SLAM scalable, and offers advantages for planning and navigation [Fernández-Moral *et al.*, 2013a; Fernández-Moral *et al.*, 2015a].

- The development of a new sensor for online acquisition of omnidirectional RGB-D images. This sensor consists of a rig of 8 Asus Xtion Pro Live sensors mounted in a radial configuration [Fernández-Moral *et al.*, 2014b]. A new SLAM approach is presented using this sensor and a hybrid metric-topological approach, based on a pose-graph of spherical RGB-D keyframes [Gokhool *et al.*, 2014].

Next, all the publications derived from this thesis are compiled:

Journals

1. *E. Fernández-Moral, J. González-Jiménez and V. Arévalo*, "Extrinsic Calibration of 2D laser rangefinders" (2014) *Submitted to:* The International Journal of Robotics Research.

Book Chapters

1. *E. Fernández-Moral, V. Arévalo and J. González-Jiménez*, "Hybrid metric - topological mapping for large scale monocular SLAM", Lecture Notes in Electrical Engineering (LNEE), 2014. Accepted for publication.

Conference Proceedings

1. *E. Fernández-Moral, W. Mayol-Cuevas, V. Arévalo and J. González-Jiménez*, "Fast place recognition with plane-based maps", Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 2013.

2. *E. Fernández-Moral, V. Arévalo and J. González-Jiménez*, "Creating metric-topological maps for large-scale monocular SLAM", Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO), Reykjavik, Iceland, 2013.

3. *T. Gokhool, M. Meilland, P. Rives and E. Fernández-Moral*, "Dense RGB-D map building from spherical RGB-D images", Proceedings of the 9th International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), Lisbon, Portugal, 2014.

4. *E. Fernández-Moral, J. González-Jiménez, P. Rives and V. Arévalo*, "Extrinsic calibration of a set of range cameras in 5 seconds without any pattern", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, USA, 2014.

5. *E. Fernández-Moral, V. Arévalo and J. González-Jiménez*, "A compact planar-patch descriptor based on color", Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO), Vienna, Austria, 2014.

1.4 Framework of this thesis

This thesis is the outcome of four years of research activity of its author as a member of the MAPIR research group[1], which is within the Department of System Engineering and Automation of the Universidad de Málaga. This research has been supported by the Spanish Government and the “European Regional Development Fund ERDF” under the contracts DPI2008-03527 and DPI2011-25483, within the framework of the projects “Construction of visual metric-topological maps for mobile robotics” and “TAROTH: New developments toward a robot at home”, respectively. The first project addresses the research of a world representation from range and visual sensors, and comprises the first two years of research of this thesis. The second one comprises the remaining of this thesis, and covers the exploitation of the previous maps for robotic applications.

During the PhD period, the author completed the doctoral program entitled “Ingeniería Mecatrónica” (Mechatronics Engineering) coordinated by the Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga. This doctoral program granted the author both a general view of the multidisciplinary field of mechatronics which combines mechanical, electrical, control and computer engineering, and more importantly a deep knowledge about mobile robotics, something that has proved fundamental throughout these years of research.

Additionally, the author complemented his academic education with the participation in the BMVA Summer School on Computer Vision (2010) at Kingston University, London. Besides he developed part of his research in collaboration with two international research groups. In 2012, he stayed 4 months with the Visual Information Laboratory, at the University of Bristol (UK), under the supervision of Dr. Walterio Mayol-Cuevas. During this period, his research was focused on exploiting planar structure for map construction and SLAM. In 2013, he stayed 9 months with the team LAGADIC, in INRIA Sophia-Antipolis (France), under the supervision of Dr. Patrick Rives. During this time, he worked on the development of an omnidirectional RGB-D device conceived for fast map construction and robot navigation.

Finally, it is worth mentioning that the scientific framework within this thesis stands in a highly competitive research area, which is driven by a growing industry of computer vision and robotic applications. In the opinion of the author, the continuous contributions in these two interrelated areas will allow the progressive integration of robots in our society.

[1] http://mapir.isa.uma.es

1.5 Structure of this thesis

The remaining chapters of this thesis are organized as follows:

Chapter 2 introduces the basic concepts regarding the calibration of different sensors, and contributes a new methodology to calibrate rigs of range sensors. The presented strategy does not require any artificial pattern since it relies on sensing planar surfaces from the environment. This calibration strategy is fast, easy to use and robust, presenting important advantages with respect to previous approaches.

Chapter 3 presents a new compact scene representation based on planes (planar patches), which can be quickly segmented from range images by region growing. This plane-based map (PbMap) is described by simple geometric and radiometric features using a graph representation. A compact colour descriptor based on the dominant colour is introduced to improve the distinctiveness of planar patches. A registration technique is also presented for continuous piecewise scene recognition. Quantitative and qualitative results are reported for place recognition, testing also the suitability for lifelong mapping.

Chapter 4 tackles the problem of hybrid metric-topological mapping. The topological structure is represented by an undirected graph where the nodes contain local metric information, and the edges define the connectivity between different local regions. This map structure is advantageous for the efficient management of the map, since only the local metric information related to the current region is used for localization and mapping. A dynamic strategy to manage the map is proposed, where local places are created depending on the interconnection between different observations. This hybrid metric-topological mapping is evaluated within both a monocular SLAM framework (using PTAM [Klein and Murray, 2007]) and with omnidirectional RGB-D SLAM introduced in chapter 5.

Chapter 5 introduces a new sensing device to capture omnidirectional RGB-D images at video rate (30 Hz) together with a SLAM solution for this kind of data. The SLAM approach is based on a metric-topological structure of keyframes, which are organized in a hierarchical network of local maps. The keyframes are described through a compact PbMap, which is used for efficient localization and loop closure. The localization obtained from PbMap registration is further refined by a dense registration technique. The consistency of the map is improved by pose-graph optimization taking into account all the keyframe connections.

Chapter 6 concludes the thesis, providing a summary of the presented research and giving an outlook of the future challenges for autonomous localization and mapping.

Chapter 2
Calibration of sensor rigs

Abstract

The operation of a robotic system requires knowing the characteristics and configuration of its sensor and motor systems. Such information is defined by the intrinsic and extrinsic parameters. The intrinsic parameters refer generally to a single sensor or actuator, describing how the measurements are modelled or how the actions are executed, respectively. On the other hand, the extrinsic parameters describe the relative poses among the sensors/actuators. Such parameters can be obtained from calibration, which is usually a prerequisite to perform other tasks as localization, mapping or navigation. This chapter reviews some relevant methods for intrinsic and extrinsic calibration, and presents several solutions for the extrinsic calibration of different combinations of range sensors developed within the work of this thesis.

2.1 Introduction

Many applications in the field of mobile robotics employ a variety of sensors, from proprioceptive sensors (like GPS, Inertial Measurement Units (IMU) or shaft encoders) to exteroceptive sensors (including vision, range or contact devices). In order to exploit efficiently the information provided by such sensorial systems, the sensors must be calibrated to: a) interpret correctly the acquired data (intrinsic calibration), and to put all the measurements in a common reference frame (extrinsic calibration). Such a calibration is also required for the robot's actuators in order to provide the right commands towards the goal. This chapter is focused on sensor calibration, though many concepts can also be applied to the calibration of actuators. For more specific literature on this the reader is referred to [Whitehouse and Culler, 2003].

The intrinsic calibration of a sensor consists of providing a model to interpret the raw measurements, so that the data is put in correspondence with world properties. Examples of intrinsic parameters are the focal length of a camera, its distortion parameters, the error model of a laser scanner, or the radius of the robot's wheels. Such parameters are usually provided by the manufacturer of the respective device, however, it may still be interesting to calibrate them in order to model deviations from the construction parameters or particular circumstances of the system (e.g. wheel inflation degree). The intrinsic calibration is required prior to the extrinsic calibration since it is needed to interpret the sensor measurements. It has been shown that computing both calibrations in a coupled manner can be helpful to reduce the errors of both [Zhang and Pless, 2004]. This section describes briefly two models employed along this thesis to calibrate the intrinsic parameters of a regular camera using a checkerboard [Zhang, 2000], and a method to calibrate the parameters of a depth camera using SLAM [Teichman *et al.*, 2013].

The extrinsic calibration among the robot's sensors (i.e. finding their relative poses) is required to exploit effectively all the sensor measurements and to perform data fusion. There exist a vast literature about this problem. These works can be classified according to the devices to be calibrated, for instance, the extrinsic calibration of regular cameras was one of the first to be investigated due to its utility for stereo vision and multiview geometry [Faugeras and Toscani, 1986]. A rig consisting of a camera and an IMU is calibrated in [Mirzaei and Roumeliotis, 2008; Guo and Roumeliotis, 2013], what is required for applications in the fields of wearable devices and aerial robotics (UAVs). Several solutions have been presented as well for the calibration of RGB cameras and laser scanners or LIDAR (e.g. [Zhang and Pless, 2004]), which have been used to build coloured point clouds of the scene [Forkuo and King, 2004]. A 3D LIDAR and a camera have been calibrated with the same purpose of registering depth and intensity information [Mirzaei *et al.*, 2012]. A Velodyne 3D LIDAR and an omnidirectional camera were calibrated in [Pandey *et al.*, 2010]. Methods for calibrating both the intrinsic and extrinsic parameters of a RGB and depth cameras have also become popular in the last years with the release of consumer RGB-D sensors like those developed by Primesense (e.g. Microsoft Kinect) [Smisek *et al.*, 2013; Herrera *et al.*, 2011].

The knowledge of the robot trajectory, and/or a map of its environment, provide valuable information to compute the calibration, and at the same time, such calibration contributes to improve the localization and mapping [Foxlin, 2002]. The problem of simultaneous calibration and localization (or SLAM) has also received considerable attention by the research community. This strategy has been applied to calibrate regular cameras [Larsen *et al.*, 1998; Heng *et al.*, 2013], laser scanners [Martinelli *et al.*, 2007], and RGB-D cameras [Teichman *et al.*, 2013; Brookshire and Teller, 2012].

Despite the large volume of literature covering different calibration problems, only a few works have addressed the extrinsic calibration between range sensors. For example, the extrinsic calibration of a set of 2D laser scanners (or laser rangefinders -LRFs-) is a problem which is present in the field of autonomous vehicles [Thrun *et al.*, 2006; Campbell *et al.*, 2010; Bohren *et al.*, 2008; Petrovskaya and Thrun, 2009; Miller *et al.*, 2011; Leonard *et al.*, 2008], where these sensors are necessary for safe navigation. But most works employing such a combination of LRFs obtain the extrinsic calibration from manual measurements or from non-general *ad-hoc* solutions, in tedious and time consuming procedures. This is a result of the difficulty to establish some kind of data association between range sensors.

A new strategy for calibrating such combinations of sensors is presented in this thesis (section 2.3), which is employed to calibrate different combinations of range sensors. The technique is based on the observation of planar surfaces at different orientations, and offers a series of advantages over previous alternatives such as its ease of use, robustness and accuracy.

2.2 Intrinsic calibration

A large body of literature can be found about the problem of intrinsic calibration of different sensors. In this section we review two methods for the intrinsic calibration of a regular RGB camera and for the depth sensor of a structure light sensor like e.g. Microsoft Kinect. These two methods are used along this thesis.

2.2.1 Calibration of a regular camera (RGB)

Finding the intrinsic parameters of a camera is a basic problem in computer vision, and it is present in many robotic applications. Such parameters describe the projection between the 3D coordinates of the scene to the 2D coordinates of the image sensor. This problem involves finding the parameters of the camera model (in general the pinhole model is used, which is defined by the focal length and principal point), and the distortion parameters (radial and tangential distortions) of the lens [Heikkila and Silvén, 1997]. The pinhole camera model is usually represented through the camera projection matrix K

$$\mathrm{K} = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.1}$$

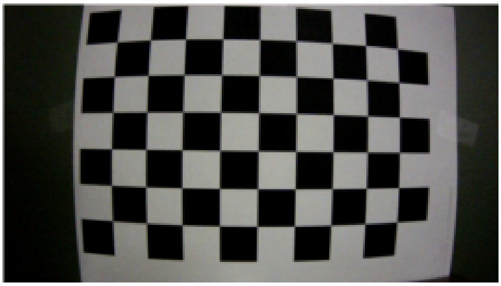


Figure 2.1: Checkerboard calibration pattern.

a 3×3 homogeneous matrix where α and β represent the scale factors (focal length) in the x and y axes, γ describes the skewness of these axes, and (u_0, v_0) are the coordinates of the principal point. Thus, the homogeneous images coordinates $\mathbf{w} = (u, v, 1)^\top$ corresponding a 3D point $\mathbf{p} = (x, y, z)^\top$ in the camera reference frame are obtained from

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathrm{K} \begin{pmatrix} x/z \\ y/z \\ 1 \end{pmatrix} \tag{2.2}$$

The lens distortion can be modelled as the combination of radial and tangential distortion. The radial distortion consists of a radially symmetric artefact induced from the lens light diffraction, while the tangential distortion is caused by misalignment of the lens or lenses. Radial distortion models are more commonly applied since its effect is generally more significant than the tangential component. The latter is not considered here, for which the reader is referred to [Devernay and Faugeras, 1995]. The radial distortion can be modelled as

$$\begin{aligned} u' &= u + (u - u_0)(k_1 r^2 + k_2 r^4 + \ldots) \\ v' &= v + (v - v_0)(k_1 r^2 + k_2 r^4 + \ldots) \end{aligned} \tag{2.3}$$

where u and v are the ideal pixel coordinates given by the pinhole model, u' and v' are the real observed pixel coordinates affected by radial distortion and r

$$r = \sqrt{(u - u_0)^2 + (v - v_0)^2} \tag{2.4}$$

is the Euclidean distance between the pixel (u, v) and the principal point. Generally, only the two first parameters k_1 and k_2 of radial distortion are computed since higher order parameters have a negligible effect.

There is a vast literature treating this problem, where a common approach is to employ a checkerboard calibration pattern (see figure 2.1) that must be observed from different orientations of the camera. Point correspondences are extracted from these observations, which are used to define geometrical equations to constrain the problem

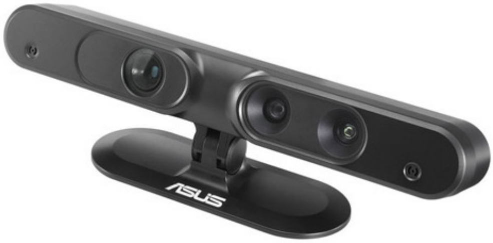


Figure 2.2: RGB-D sensor: Asus Xtion Pro Live.

[Zhang, 2000]. This strategy has been applied along this thesis to find the intrinsic parameters of different RGB cameras. Concretely, it is used in sections 4.4 and 5.2.1. The method employed here for calibration is publicly available[1] within the project MRPT [Blanco, 2008].

2.2.2 Calibration of depth cameras

Depth cameras are increasingly popular in mobile robotics thanks to the arrival of low-cost sensors like Asus Xtion Pro. Some other technologies for depth imaging include time-of-flight (ToF) cameras and 3D LIDAR, with considerable differences in price and working conditions. In this section, we focus on the calibration of structured-light cameras like Asus Xtion Pro or Microsoft Kinect since these sensors are used along this thesis. For that, we outline some relevant works and explain in more detail the one followed here.

Structured-light sensors are composed of an infrared (IR) camera and an IR projector (see figure 2.2). The depth image is obtained from stereo matching of the infrared projected pattern, and the specifications of this process are unknown to the user. Thus, a model for the depth image formation like the one of the previous section for regular intensity cameras is not available. As a consequence, many works that address the calibration of RGB-D sensors solve for the intrinsic parameters of RGB and IR cameras and for the relative pose of these, but do not deal with the intrinsic parameters of depth imaging [Herrera *et al.*, 2011]. More recent works have addressed the intrinsic calibration of this type of depth sensors proposing to treat each pixel (in fact, regions of pixels) individually, to identify the bias of them through statistic methods which imply the observation of a static scene using SLAM [Teichman *et al.*, 2013], or the observation of a checkerboard pattern [Basso *et al.*, 2014b]. In this thesis we employ the first of this methods to calibrate the depth provided by the RGB-D sensors used along this thesis.

The depth error of RGB-D sensors from PrimeSense (including Kinect and Asus Xtion) increases with distance, introducing a bias in the measurements. Such a bias is evident when we see the deformation of a flat surface as it is observed from increas-

[1] `http://www.mrpt.org/list-of-mrpt-apps/application_camera-calib/`

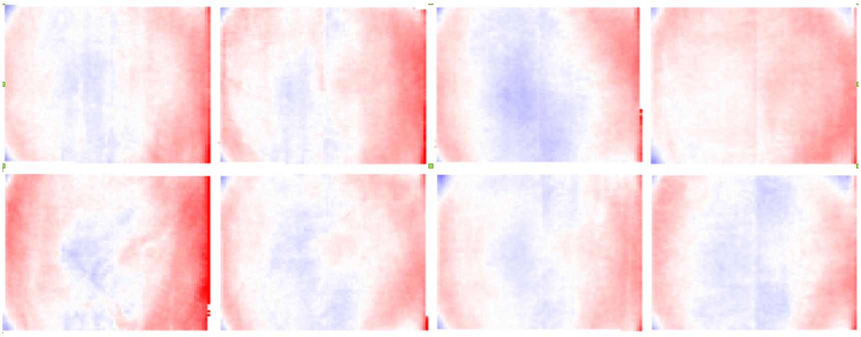


Figure 2.3: Intrinsic calibration. Image of multipliers for the 8 different sensors at the reference depth of 4 m.

ing distance. In [Teichman *et al.*, 2013], the authors propose to model the intrinsic parameters using a discrete image of multipliers, where every pixel is updated according to its coordinates and depth. This solution, which is publicly available[2], has been applied in this thesis to estimate the intrinsic parameters of different RGB-D sensors employed in our experiments. This technique was chosen because it was the only one which models the bias in the depth measurements. A sample of the resulting images of multipliers for different sensors Asus Xtion Pro Live is shown in figure 2.3, where we can see that the bias of each sensor is different even though they are the same sensor type.

2.3 Extrinsic calibration of range sensors

The extrinsic calibration of different sensors is of very practical interest in robotics. This problem has been widely studied and different solutions have been presented for a variety of sensor configurations [Zhang and Pless, 2004; Le and Ng, 2009; Ha, 2012; Heng *et al.*, 2013; Schneider *et al.*, 2013]. The case of extrinsic calibration of range sensors is a problem with fewer results in the literature, where only solutions to very specific problems have been presented. The reasons for that are the difficulty to establish some kind of data association between range sensors in arbitrary orientations, and the high cost of many of these sensors, though this cost is being reduced as new technological advances appear [AsusXPL, 2011].

A new strategy to calibrate a combination of range sensors is proposed in this section. The calibration methods proposed here are all based on the observation of planar surfaces at different orientations to establish constraints on the sensor relative poses. The calibration problem is tackled as a maximum likelihood estimation (MLE) for the graph of constraints inferred between the different sensors. This formulation permits to solve the calibration of different types of sensors, as each measurement

[2] http://cs.stanford.edu/people/teichman/octo/clams/

is weighted according to its uncertainty. The uncertainty of the resulting calibration can also be estimated, what is useful for SLAM and to evaluate the precision of the calibration. The observability of these problems is analysed through the Fisher information Matrix [Van Trees and Bell, 2007], presenting minimal solutions which only require a single observation of the sensors.

The calibration of different combination of range sensors is dealt with in this chapter: section 2.3.1 tackles the extrinsic calibration of a set of 2D LRFs; the calibration of a set of range cameras is addressed in section 2.3.2; and finally, the calibration between range cameras and 2D LRFs is presented in section 2.3.3. The experimental results confirm the efficacy and robustness of these calibration methods. These three problems are solved separately as they are stated upon different constraints, thus having different observability and convergence conditions.

2.3.1 Calibration of several 2D laser rangefinders

The extrinsic calibration of several 2D laser rangefinders or LIDAR is of very practical interest for autonomous vehicles and for mobile robotics. Combinations of LRFs have been employed for 3D mapping in outdoor [Borrmann *et al.*, 2008; Barber *et al.*, 2008; Haala *et al.*, 2008] and indoor environments [Thrun *et al.*, 2000], and also for safe navigation [Victorino *et al.*, 2003]. This calibration problem becomes more relevant with the advent of autonomous cars [Thrun *et al.*, 2006; Campbell *et al.*, 2010; Bohren *et al.*, 2008; Petrovskaya and Thrun, 2009; Miller *et al.*, 2011; Leonard *et al.*, 2008], where the information provided by such sensors is essential to avoid possible collisions.

This section presents a novel solution for the general problem of extrinsic calibration of 2D LRFs, which is based on the observation of perpendicular planes from any structured scene (i.e. Manhattan like world). Then, the calibration is computed by imposing co-planarity and perpendicularity constraints on the line segments extracted by the different laser scanners. No external information in the form of calibration patterns or auxiliary sensors is required. Only a rough approximation of the sensor relative poses must be provided, which can be guessed from simple visual inspection of the rig. This method can be used to calibrate any set of rigidly joined LRFs where there are at least two sensors with non-parallel scanning planes. The flexibility of our method permits its application to different problems. For example, it can be used to re-calibrate the LRFs mounted on an autonomous car, where the sensors relative poses may change over time as a result of the vehicle vibrations [Bohren *et al.*, 2008].

2.3.1.1 Related works

Among the robotic systems found in the literature that employ a combination of LRFs, only a few of them report a calibration technique [Huang *et al.*, 2010; Blanco *et al.*, 2009b; Gao and Spletzer, 2010]. Many other works like [Thrun *et al.*, 2006; Miller *et al.*, 2011; Campbell *et al.*, 2010; Bohren *et al.*, 2008; Petrovskaya and Thrun, 2009], do not report any calibration process, so, it is reasonable to suppose that they obtain the sensor's relative poses from manual measurements on their setups, like in [Blanco-Claraco *et al.*, 2014]. Such procedures are prone to errors that may severely affect the performance of mapping and navigation methods, especially when the laser scanners have a long working range, so that small rotation errors can produce significant distortions in the map [Miller *et al.*, 2011; Leonard *et al.*, 2008]. Apart from the limitations in accuracy, measuring the sensors relative poses by hand is also tedious and time consuming.

Generally, the preferred strategy to calibrate exteroceptive sensors is to use their own measurements to establish some kind of data association between their observations. The extrinsic calibration of 2D range scanners in arbitrary poses proves to be more difficult than for RGB or depth cameras, since distinctive features are significantly more scarce in the first. This calibration strategy has been demonstrated for LRFs in some particular problems, but the solutions reported share one or more of the next limitations: they need supervised data association in controlled conditions; they need external information (extra sensors, a pattern or landmarks placed manually in the environment); or they are specific for a particular configuration of the sensor rig. For example, vertical posts of traffic signs are segmented and matched in a supervised way in [Huang *et al.*, 2010]. In [Gao and Spletzer, 2010], a solution is presented based on the matching of reflecting landmarks that are manually placed in the environment. Without using particular targets, calibration is achieved in [Blanco *et al.*, 2009b] by making use of the vehicle's odometry to maximize the fitting of the 3D point clouds built from the different LRFs, what requires extra sensors (cameras or high precision GPS) to improve the accuracy of the vehicle's odometry. Another approach consists of matching the trajectories of dynamic objects (or people) in the scene [Glas *et al.*, 2010; Schenk *et al.*, 2012]. For that, the trajectory of one, or several objects, is tracked independently by each LRF, and these trajectories are registered to constraint the sensor's relative poses. This solution is suitable for static systems where all the LRFs scan a common space (nearly in the same plane), but like the approaches above, it is not valid to calibrate LRFs in arbitrary poses. In contrast to those works, a general method to calibrate LRFs like the one proposed in this thesis, is useful to deal with many robot and autonomous vehicle configurations.

Ego-motion approaches have also been exploited to calibrate different combinations of sensors. For instance, visual and range odometry [Brookshire and Teller, 2012; Heng *et al.*, 2013; Schneider *et al.*, 2013] have been employed to minimize the fitting error of the independently estimated sensor trajectories. Also, ego-motion has been used in combination with wheel odometry to determine the intrinsic parameters of the odometry together with the relative pose of a laser sensor with respect to the robot's frame [Censi *et al.*, 2013]. However, this strategy is only applicable when the laser scanner moves in its own plane of measurement (typically, planar movement of a vehicle with an horizontal LRF), otherwise the laser ego-motion cannot be estimated. This last work addresses a different, but complementary problem to the one tackled here. Therefore, a combination of this technique with the one proposed here would be interesting in many problems to calibrate several LRFs with respect to the vehicle's frame.

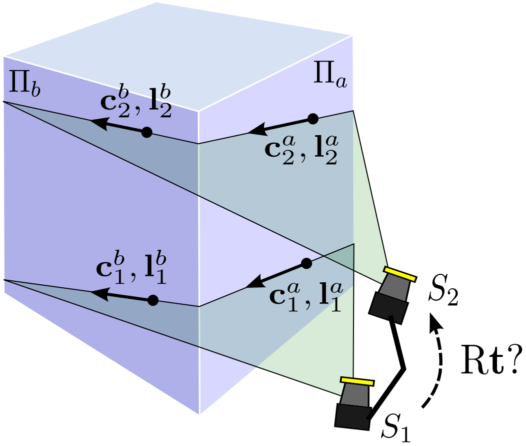

Π_b
Π_a
$\mathbf{c}_2^b, \mathbf{l}_2^b$
$\mathbf{c}_2^a, \mathbf{l}_2^a$
$\mathbf{c}_1^b, \mathbf{l}_1^b$
$\mathbf{c}_1^a, \mathbf{l}_1^a$
S_2
Rt?
S_1

Figure 2.4: Observation of a corner structure by a rig with two LRFs.

Contribution

This section presents, to the best of our knowledge, the first general approach to calibrate a set of LRFs in arbitrary positions. This solution only requires observing perpendicular planes from any structured scene. The observability and the convergence of the method are studied. A C++ implementation of this method is also provided together with a testing dataset[34].

In comparison to previous approaches, our method is applicable to almost any geometric configuration of the sensors (with at least two non-parallel sensors); it does not require auxiliary sensors or a calibration pattern; it is accurate and fast, indeed, the calibration can be achieved from a single observation; and finally, our method provides an estimation of the calibration uncertainty.

In the following section we describe the calibration method using constraint equations derived from the co-planarity and perpendicularity of the observed planes (section 2.3.1.2). Section 2.3.1.3 addresses the optimization problem stated upon these constraints within a probabilistic framework that takes into account the precision of the sensor measurements. We analyse the observability conditions (section 2.3.1.4) and study the convergence region for the solution (section 2.3.1.5). Experimental results are presented to validate our approach with both simulated and real data (section 2.3.1.6). Finally, the results are discussed and the conclusions are outlined.

2.3.1.2 Calibration approach

Our proposal for finding the extrinsic calibration (i.e. relative poses) between two or more LRFs relies on establishing geometric constraints from the simultaneous observation of pairs of perpendicular planes (see figure 2.4). For readability, let us define a corner as a pair of perpendicular planar surfaces, not necessarily intersecting[5]. Then, the geometric constraints are inferred from: 1) the co-planarity of the observed line segments lying on each face of the corner, and 2) the perpendicularity of both planar surfaces.

A minimum of two corner observations from different orientations are required to calibrate a pair of LRFs (note that these can be obtained from a single observation of the rig when three perpendicular planes are visible, as it is shown in figure 2.7). However, it is preferable to take more observations to compensate for the noise in the measurements, increasing the accuracy of the calibration. One of the easiest ways to obtain such observations is by rotating the sensor rig in front of a corner. From these observations, the calibration problem is stated as the optimization of a graph of constraints, similar to the problem of pose graph SLAM [Grisetti *et al.*, 2010].

Before going into the calibration method itself, we address related issues like how the lines are represented and segmented from the laser scans, how to detect corner observations and how to derive constraints on the sensor relative poses from them.

Line representation and segmentation

The planar structures of the environment are sampled by the LRFs as line segments. These lines can be extracted from the scans provided by each LRF in a number of ways [Nguyen *et al.*, 2005]. Here we have implemented a segmentation method based on RANSAC [Fischler and Bolles, 1981], although other approaches like those based on region growing [Borges and Aldon, 2000] or on Hough transform [Forsberg *et al.*, 1995] may also be applied. The RANSAC procedure searches for the parameters $\{A,B,C\}$ of a 2D line which maximize the number of points $\mathbf{p}_i = (x_i, y_i)^\top$ supporting the model

$$Ax_i + By_i + C \leq \varepsilon \tag{2.5}$$

being ε a threshold used to differentiate between inliers and outliers. An advantage of using RANSAC is that unconnected collinear segments are automatically clustered as the same line, simplifying the subsequent optimization process.

The segmented lines are represented in 2D in the LRF's reference system by the normalized direction vector $l = (l_x, l_y)^\top$ and an arbitrary point of the line (see figure 2.5). For such a point we have chosen the centroid of the line's inliers $c = (c_x, c_y)^\top$

[3] `https://github.com/EduFdez/mrpt/tree/LRF-calib/apps/LRF-calib/`

[4] `https://github.com/EduFdez/mrpt/blob/LRF-calib/share/mrpt/datasets/3LRFs_dataset_demo.rawlog`

[5] This procedure of calibration can make use of pairs of oblique (non-parallel) planes, not requiring perpendicularity. However, by considering only perpendicular corners we facilitate both the readability of this section and its implementation without adding any strong limitation, since any structured scene (i.e. man-made environment) contains perpendicular planes.

since it has less uncertainty to belong to the line than any measured point. These parameters and their covariances are estimated assuming unbiased, identically distributed (i.i.d.) Gaussian noise in the LRF measurements. The covariances are used latter as a measure of the uncertainty of the observed lines. In the literature, it is often assumed a model where the noise only affects the range measurements, with exact bearing directions [Arras and Siegwart, 1998; Diosi and Kleeman, 2003]. However, such a model introduces linearization errors that produce biased estimates of the line parameters. To avoid this, a common approach (see [Arras and Siegwart, 1998; Diosi and Kleeman, 2003]) that we follow here is to approximate the covariance of each point $\mathbf{p}_i$ in Euclidean coordinates as $\Sigma_{p_i} = \sigma_i^2 I$. Then, the maximum likelihood estimation of the centroid $\mathbf{c}$ is calculated as

$$\mathbf{c} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i \tag{2.6}$$

and its covariance Σ_c is given by

$$\Sigma_c = \begin{bmatrix} \frac{\sigma^2}{N} & 0 \\ 0 & \frac{\sigma^2}{N} \end{bmatrix} \tag{2.7}$$

The line direction vector $\mathbf{l}$ is obtained as the eigenvector corresponding to the largest eigenvalue of the point dispersion matrix M

$$M = \sum_{i=1}^{N}(\mathbf{p}_i - \mathbf{c})(\mathbf{p}_i - \mathbf{c})^\top \tag{2.8}$$

and its covariance

$$\Sigma_l = H^+ \tag{2.9}$$

is calculated following [Pathak *et al.*, 2010c] as the Moore-Penrose generalized inverse of

$$H = \frac{1}{\sigma^2}\sum_{i=1}^{N}\begin{bmatrix} y_i^2 & -y_i x_i \\ -x_i y_i & x_i^2 \end{bmatrix} \tag{2.10}$$

In the rest of the section the lines are represented in 3D in the LRF reference system (see figure 2.5) by setting the z component of the line parameters and their covariances to zero.

Corner constraints

Let S_1 and S_2 be two rigidly jointed LRFs, each one observing two line segments $\{L_1^a, L_1^b\}$ and $\{L_2^a, L_2^b\}$ from two perpendicular planes Π_a and Π_b. Let $[\mathrm{R}_1|\mathbf{t}_1], [\mathrm{R}_2|\mathbf{t}_2] \in \mathbb{SE}(3)$ be the LRFs poses with respect to a common coordinate system, with the rotations $\mathrm{R} \in \mathbb{SO}(3)$ represented as 3×3 matrices and the translations $\mathbf{t} \in \mathbb{R}^3$.

Co-planarity constraint: A co-planarity constraint is inferred for the lines segmented by two LRFs that observe the same plane, for instance, the lines $\{L_1^a, L_2^a\}$ on the plane

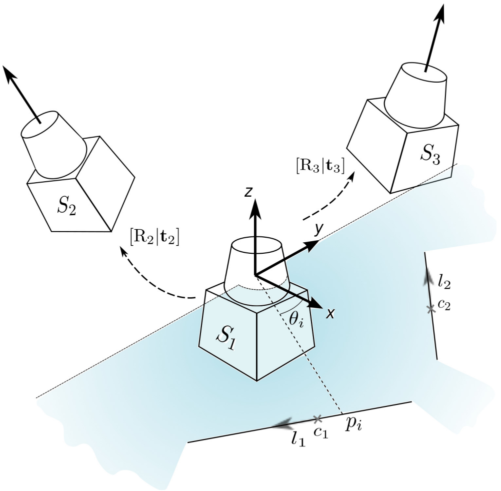


Figure 2.5: A rig of 3 LRFs in different orientations (i.e. the z axis of each LRF are linearly independent).

Π_a in figure 2.4. Both line direction vectors and the vector joining any two points $\mathbf{p}_1^a$ and $\mathbf{p}_2^a$ of the observed lines, all referred to the same coordinate system, form a matrix of deficient rank as they are all parallel to the plane Π_a. Therefore, the determinant of that matrix, which is equal to the mixed triple product of the 3 vectors, must be zero. This is expressed as

$$(\mathrm{R}_1\mathbf{l}_1^a \times \mathrm{R}_2\mathbf{l}_2^a) \cdot (\mathrm{R}_1\mathbf{p}_1^a + \mathbf{t}_1 - \mathrm{R}_2\mathbf{p}_2^a - \mathbf{t}_2) = 0 \tag{2.11}$$

This condition can also be interpreted as a statement for the perpendicularity between the plane's normal vector $\mathbf{n}^a = \mathrm{R}_1\mathbf{l}_1^a \times \mathrm{R}_2\mathbf{l}_2^a$ and any vector joining a pair of points from the scanned lines. For such points, we make use of the centroids $\mathbf{c}_1^a$ and $\mathbf{c}_2^a$, simplifying the optimization procedure presented later on.

Orthogonality constraint: Another constraint can be stated for the relative rotation of a pair of LRFs that observe a corner defined by the perpendicular planes Π_a and Π_b, so that

$$\mathbf{n}^a \cdot \mathbf{n}^b = (\mathrm{R}_1\mathbf{l}_1^a \times \mathrm{R}_2\mathbf{l}_2^a) \cdot (\mathrm{R}_1\mathbf{l}_1^b \times \mathrm{R}_2\mathbf{l}_2^b) = 0 \tag{2.12}$$

Notice that the first constraint affects only the observation of a plane, while the second one implies the observation of a corner. Regarding the calibration problem, the former involves the relative rotation and translation of the sensors, while the latter affects only the rotation.

Corner detection

Let us call CO the corner observed by a pair of rigidly jointed LRFs. Specifically, a CO is described by the set: $CO = \{j, L_j^a, L_j^b, k, L_k^a, L_k^b\}$, where j and k are the indices of the LRFs, a and b refer to the respective corner faces (orthogonal planes), and the lines are represented by $L_j^a = \{\mathbf{c}_j^a, \mathbf{l}_j^a, \Sigma_{c_j}^a, \Sigma_{l_j}^a\}$.

Detecting a CO is not trivial when the relative poses of the sensors are not known, as it is the case here. Indeed, since the information provided by the LRF measurements is purely geometric, we can only know that two pairs of lines observed by two LRFs come from a corner once we know the sensor's calibration. Thus, the resulting problem implies detecting the COs and estimating the calibration simultaneously. This can be tackled in a hypothesize-and-test framework, where many potential COs are generated from the rig observations by grouping sets of two pairs of lines seen by a pair of LRFs, where some of them must come from real corners.

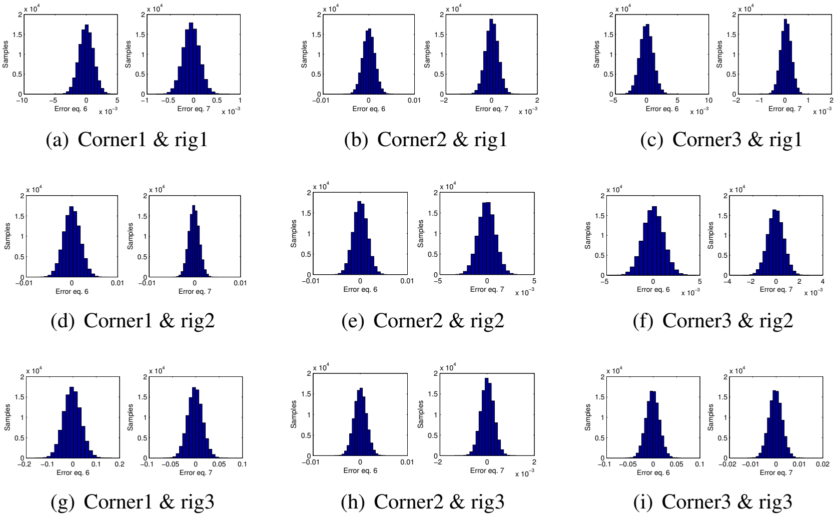
After a number of observations are taken from different poses of the sensor rig, inconsistent COs are ruled out robustly using RANSAC [Fischler and Bolles, 1981] taking into account the restrictions in equations (2.11-2.12). For that, a candidate extrinsic calibration is calculated from a minimum set of randomly selected COs (as explained in the next section). Then, the number of consistent COs for such calibration is evaluated. This process is repeated iteratively searching for the maximum consensus in COs. The result of this process is the largest set of consistent COs (inliers) and the calibration computed from them. Empirically, we have verified that the correct calibration always corresponds to the largest number of inliers, even when the number of outliers is considerably larger than the number of inliers ($n_{inliers}/n_{outliers} \sim 0.1$). This situation where the number of outliers is much larger than that of inliers results in a slow calibration process, however, it is not critical since this can be done offline.

It is worth mentioning that if we have a rough knowledge of the sensors relative poses, what is very common in practice, some constraints can be set for the selection of COs so that fewer outliers are selected at a first instance. Such restrictions are not applied here for the sake of generality.

The process to obtain the COs can perform automatically from the streaming data of the sensors (see for example the video at `http://youtu.be/YG4ShgyIUHQ`). Errors derived from the motion of the sensor rig are neglected here since the rig's velocity is small with respect to the acquisition time. In order to decide when should we stop gathering COs, a convergence condition for the maximum uncertainty (covariance) of the resulting calibration can be set to stop this process automatically [Fernández-Moral *et al.*, 2014b].

Gaussian error model

It is common to assume Gaussian error models in robotics. This section verifies such assumption for the error models employed here. In the simulation experiments carried out in this work, the depth measurements are generated with additive Gaussian noise of zero mean, and no error in the bearing. Although, the covariances of the



(a) Corner1 & rig1 (b) Corner2 & rig1 (c) Corner3 & rig1

(d) Corner1 & rig2 (e) Corner2 & rig2 (f) Corner3 & rig2

(g) Corner1 & rig3 (h) Corner2 & rig3 (i) Corner3 & rig3

Figure 2.6: Monte Carlo simulation of a corner observation at three different orientations (a different corner is shown in each column) with three different rigs of two LRFs (a different rig is shown in each row). The planarity and perpendicularity errors are shown at the left and right of each graph, respectively.

measured points are assigned as diagonal matrices (eq. 2.7) in Euclidean coordinates (a common practice in most of the related works [Arras and Siegwart, 1998; Diosi and Kleeman, 2003]). Moreover, several operations are applied to Gaussian random variables in the error model, so that the resulting distribution is not necessarily Gaussian. In this section, we conduct some Monte Carlo simulation with 10^5 samples, to check how are distributed the planarity error (eq. 2.11) and the perpendicularity error (eq. 2.12) of a corner observation for different normal errors in the measurements. These errors are simulated for three different corners at different orientations, and for three rig configurations with two LRFs (nine evaluations are shown). The results of these simulations are shown in figure 2.6, confirming that both errors follow normal distributions of zero mean. This test has been repeated for different resolutions of the LRFs, obtaining similar results.

2.3.1.3 Problem formulation

Given a set of COs gathered from different orientations of a rig consisting of m LRFs $\{S_1,...,S_j,...,S_m\}$ where, without loss of generality, the sensor S_1 is chosen as the reference coordinate system and each LRF S_j is located with a relative transformation $[\mathrm{R}_j|\mathbf{t}_j] \in \mathbb{SE}(3)$ with respect to S_1. Then, we want to estimate the optimal $\{\mathrm{R},\mathbf{t}\} = \{[\mathrm{R}_2|\mathbf{t}_2],...,[\mathrm{R}_j|\mathbf{t}_j],...,[\mathrm{R}_m|\mathbf{t}_m]\}$ that minimize the errors of the constraints in eqs. 2.11-2.12, assuming independence between the *CO*s and measurements affected by unbiased Gaussian noise as modelled in section 2.3.1.2.

Maximum likelihood estimation

The above problem is formulated as the maximum likelihood estimation (MLE) of the relative poses $\{\mathrm{R}, \mathbf{t}\}$ for the given *CO*s, which is calculated from the maximization of the log-likelihood

$$\underset{\{\mathrm{R},\mathbf{t}\}}{\operatorname{argmax}} \left(\ln \prod_{i=1}^{N} p(CO_i | \{\mathrm{R}, \mathbf{t}\}) \right) \tag{2.13}$$

where the likelihood of $\{\mathrm{R}, \mathbf{t}\}$ for the i-th CO is calculated from the constraints presented above (eqs. 2.11-2.12). It is expressed as the multiplication of the likelihood for each constraint, two co-planarity constraints from eq. 2.11 (one per plane in the corner) that affect the estimate of both rotation and translation and one constraint from eq. 2.12 that affects only the estimate of the relative rotation. Thus, for a given CO_i observed by the LRFs j and k, we have:

$$p(CO_i | \{\mathrm{R}, \mathbf{t}\}) = \\ p(CO_i^a | \mathrm{R}_j, \mathbf{t}_j, \mathrm{R}_k, \mathbf{t}_k) \cdot p(CO_i^b | \mathrm{R}_j, \mathbf{t}_j, \mathrm{R}_k, \mathbf{t}_k) \cdot p(CO_i^{ab} | \mathrm{R}_j, \mathrm{R}_k) \tag{2.14}$$

with the superindices a and b referring to the co-planarity constraints inferred from each plane, and the superindex ab referring to the perpendicularity condition. As verified experimentally through the Montecarlo simulations above, the above probabilities follow Gaussian distributions. Concretely, the first two elements of the right term in eq. 2.14 are given by

$$p(CO_i^a | \mathrm{R}_j, \mathbf{t}_j, \mathrm{R}_k, \mathbf{t}_k) = \frac{1}{\sqrt{2\pi}\sigma_i^a} \exp\left(-\frac{1}{2} \frac{r_i^2}{(\sigma_i^a)^2} \right) = \\ \frac{1}{\sqrt{2\pi}\sigma_i^a} \exp\left(-\frac{(\mathbf{n}_{jk}^a \cdot (\mathrm{R}_j \mathbf{c}_j^a + \mathbf{t}_j - \mathrm{R}_k \mathbf{c}_k^a - \mathbf{t}_k))^2}{2(\sigma_i^a)^2} \right) \tag{2.15}$$

with

$$\mathbf{n}_{jk}^a = \mathrm{R}_j \mathbf{l}_j^a \times \mathrm{R}_k \mathbf{l}_k^a \tag{2.16}$$

being σ_i^a the standard deviation of the residual r_i which is computed from eq. 2.11. The expression of the probability from the co-planarity constraint of the plane Π_b is the same with the exception of the superindices. On the other hand, the probability inferred from eq. 2.12 is given by

$$p(CO_i^{ab} | \mathrm{R}_j, \mathrm{R}_k) = \frac{1}{\sqrt{2\pi}\sigma_i^{ab}} \exp\left(-\frac{(\mathbf{n}_{jk}^a \cdot \mathbf{n}_{jk}^b)^2}{2(\sigma_i^{ab})^2} \right) \tag{2.17}$$

being σ_i^{ab} the standard deviation of the error of (eq. 2.12). Both standard deviations (σ_i^a and σ_i^{ab}) are computed through linearisation from a first order Taylor approximation of the error functions. Their derivation is detailed in the appendix C.

When these standard deviations are constant with respect to the model parameters, the solution of the MLE in (2.13) coincides with that of the weighted, non-linear least squares problem expressed as

$$\begin{aligned}\underset{\{\mathrm{R},\mathbf{t}\}}{\operatorname{argmin}} \sum_{i=1}^{N} \Big(& \omega_i^a (\mathbf{n}_{jk}^a \cdot (\mathrm{R}_j \mathbf{c}_j^a + \mathbf{t}_j - \mathrm{R}_k \mathbf{c}_k^a - \mathbf{t}_k))^2 + \\ & \omega_i^b (\mathbf{n}_{jk}^b \cdot (\mathrm{R}_j \mathbf{c}_j^b + \mathbf{t}_j - \mathrm{R}_k \mathbf{c}_k^b - \mathbf{t}_k))^2 + \\ & \omega_i^{ab} (\mathbf{n}_{jk}^a \cdot \mathbf{n}_{jk}^b)^2 \Big) \end{aligned} \tag{2.18}$$

where ω_i^x (the superindex x stands for a, b or ab) is the weight of the corresponding residual from CO_i

$$\omega_i^x = \frac{1}{(\sigma_i^x)^2} \tag{2.19}$$

This problem is reformulated using Lie algebra (see appendix B) to represent the poses with a minimal parametrization on a manifold. For that, the rotations are represented as the composition of a guessed rotation and a rotation increment represented with the exponential map ($e^{\mu_j}\mathrm{R}_j$), with the rotation increment $e^{\mu_j} \in \mathbb{SO}(3)$. The translations are also represented as the sum of a guessed translation plus an increment ($\mathbf{t}_j + \Delta\mathbf{t}_j$), both in $\mathbb{R}^3$. The resulting non-linear least squares problem is solved iteratively using Levenberg-Marquardt

$$[\mu_2^k, \Delta\mathbf{t}_2^k, ..., \mu_m^k, \Delta\mathbf{t}_m^k]^\top = -(H + \lambda\, diag(H))^{-1} g \tag{2.20}$$

being λ the Levenberg-Marquardt's damping factor. H is the Hessian (a symmetric matrix of dimension $6(m-1)$) and g is the Gradient (a column vector of dimension $6(m-1)$) of the cost function, which are calculated as

$$H = \sum_{i=1}^{N} J_i^\top \omega_i J_i \;, \;\; g = \sum_{i=1}^{N} J_i^\top \omega_i r_i \tag{2.21}$$

where N is the number of constraints of this optimization, being r_i the residual defined above and J_i the Jacobian for each constraint (remember that each CO provides three constraints). The Jacobian J_i^a corresponding to the constraint from eq. 2.11 is calculated as

$$\begin{aligned} J_i^a = [...; & \underbrace{(\mathbf{l}_j^{aW} \times (\mathbf{l}_k^{aW} \times (\mathbf{c}_j^{aW} - \mathbf{c}_k^{aW})) + \mathrm{R}_j \mathbf{c}_j^a \times \mathbf{n}_{jk}^a}_{J_{\mu_j}};\ \underbrace{\mathbf{n}_{jk}^a}_{J_{\Delta\mathbf{t}_j}};...; \\ ...; & \underbrace{-(\mathbf{l}_k^{aW} \times (\mathbf{l}_j^{aW} \times (\mathbf{c}_j^{aW} - \mathbf{c}_k^{aW})) - \mathrm{R}_k \mathbf{c}_k^a \times \mathbf{n}_{jk}^a}_{J_{\mu_k}}; \underbrace{-\mathbf{n}_{jk}^a}_{J_{\Delta\mathbf{t}_k}};...]^\top \end{aligned} \tag{2.22}$$

where the superindex W refers to the common system of coordinates, so that[6]

$$l_j^{aW} = \mathrm{R}_j \mathbf{l}_j^a \; , \; l_k^{aW} = \mathrm{R}_k \mathbf{l}_k^a$$

$$\mathbf{c}_j^{aW} = \mathrm{R}_j \mathbf{c}_j^a + \mathbf{t}_j \; , \; \mathbf{c}_k^{aW} = \mathrm{R}_k \mathbf{c}_k^a + \mathbf{t}_k$$

This Jacobian (a row vector of dimension $6(m-1)$) contains four blocks of 1×3 vectors corresponding to the derivatives of the residual with respect to μ_j, $\Delta\mathbf{t}_j$ and μ_k, $\Delta\mathbf{t}_k$, respectively.

On the other hand, the Jacobian J_i^{ab} of the residual from eq. 2.12 is given by

$$\begin{aligned} J_i^{ab} = [...; & \underbrace{\mathbf{l}_j^{bW} \times (\mathbf{l}_k^{bW} \times \mathbf{n}_{jk}^a) + \mathbf{l}_j^{aW} \times (\mathbf{l}_k^{aW} \times \mathbf{n}_{jk}^b)}_{J_{\mu_j}}; ... \\ ...; & \underbrace{-\mathbf{l}_k^{bW} \times (\mathbf{l}_j^{bW} \times \mathbf{n}_{jk}^a) - \mathbf{l}_k^{aW} \times (\mathbf{l}_j^{aW} \times \mathbf{n}_{jk}^b)}_{J_{\mu_k}}; ...]^\top \end{aligned} \tag{2.23}$$

The two blocks of 1×3 vectors of this Jacobian correspond to the derivatives of the residual with respect to μ_j and μ_k respectively, the blocks corresponding to the rest of elements in $\{\mathrm{R}, \mathbf{t}\}$ being zero.

This optimization is solved iteratively

$$\mathrm{R}_j^{k+1} = e^{\mu_j^k} \mathrm{R}_j^k \; , \;\; \mathbf{t}_j^{k+1} = \Delta\mathbf{t}_j^k + \mathbf{t}_j^k \; , \;\; j \in [2, m] \tag{2.24}$$

from an initial guess for the sensor relative poses, which may be obtained from a rough measurement of the rig. Once the problem is solved, the covariance of the resulting calibration is calculated as the inverse of the Hessian of the cost function in (2.18) [Fernández-Madrigal and Claraco, 2013]. For a more detailed derivation of Maximum Likelihood Estimation and Least Squares the reader is referred to appendix A.

2.3.1.4 Observability

The problem of estimating the relative poses has $6(m-1)$ degrees of freedom (DoF), with m being the number of LRFs. From the formulation presented in the previous section, we have seen that each *CO* leads to three constraints for the relative poses of the corresponding pair of LRFs. Therefore, at least $2(m-1)$ *CO*s are needed to provide as many equations as unknowns for solving the problem.

We are also interested in knowing how these observations should be taken in order to provide the necessary information to solve the calibration. The analysis of the observability of calibration problems provides valuable information about the procedure to gather such data [Martinelli, 2011; Censi *et al.*, 2013]. Such analysis is carried out here by studying the rank of the Fisher Information Matrix (FIM) of the estimation problem(see appendix D). The key concept here is that when the FIM is singular,

[6] For clarity, the *CO* index i will be omitted in subsequent operations that affect only to the same *CO*.

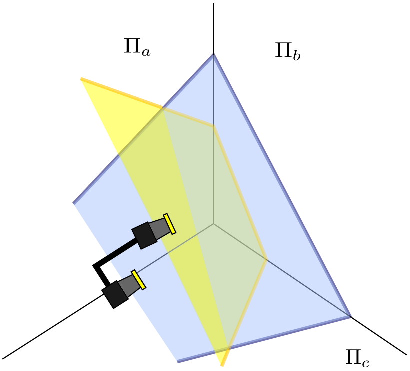


Figure 2.7: Observation of a corner with three perpendicular planes by two LRFs.

the information carried by the data (observations) is not sufficient and the problem is under-constrained ($\{\mathrm{R}|\mathbf{t}\}$ is unobservable). We wish to identify these situations of unobservability in order to avoid them in practice.

The FIM can be expressed in matrix notation by transforming the sum term in (2.21) to

$$FIM = J^\top \Omega J \tag{2.25}$$

where J is a matrix concatenating the Jacobians J_i of the residuals, and Ω is a diagonal matrix containing the weights ω_i of such residuals. Since Ω is diagonal with all the elements being positive, the rank of FIM is the same as the rank of J

$$rank(FIM) = rank(J^\top J) = rank(J) \tag{2.26}$$

Therefore, the problem has a solution when

$$rank(J) = 6(m-1) \tag{2.27}$$

By analysing the structure of the different Jacobians J_i for each CO, we can notice that a CO provides three linearly independent rows for J when a corner is observed in a new orientation (eqs. 2.22 and 2.23). To get a deeper insight into this, consider a block of the Jacobian in eq. 2.23, for instance J_{μ_j}. Each corner observation in a new, linearly independent direction, expressed by $\mathbf{n}^a_{jk} \times \mathbf{n}^b_{jk}$, contributes to constrain the problem for μ_j. For the Jacobian in eq. 2.22, it can be verified that each plane observation providing a linearly independent $\mathbf{n}_{jk}$ results in a linearly independent J_i which constrains both the relative rotation and the relative translation between the sensors S_j and S_k. Therefore, two observations of a corner from different orientations suffice to solve the calibration of a pair of sensors. Moreover, a single observation of a corner with three perpendicular planes (as shown in figure 2.7), provides enough information to solve the problem since it contains already 3 independent normal vectors of the plane, and 3 independent orthogonal constraints.

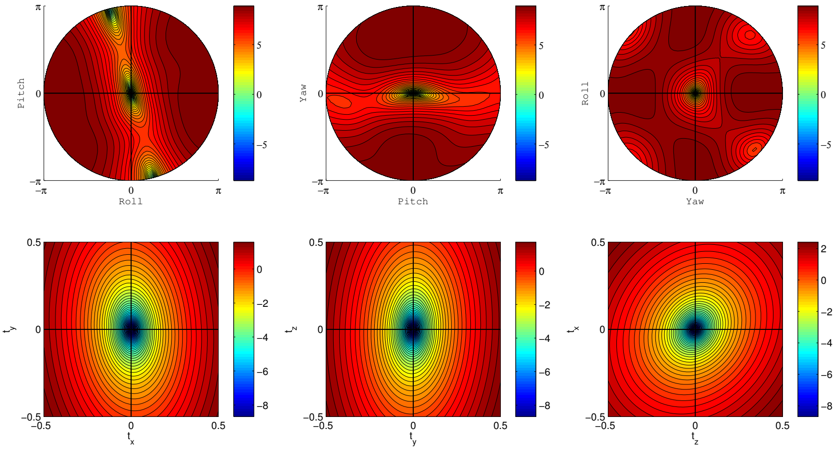


Figure 2.8: Error maps of the calibration of a pair of LRFs for 6 different combinations of 2 DoFs using 100 COs. The configuration of the rig corresponds to the one depicted in figure 2.12 for the sensors S_1 and S_2. The heat maps show the residual error from eq. 2.18 in logarithmic scale, with the correct calibration at the center of each graph, lying on a local minimum. The free DoFs in rotation take all possible values in the domain $\tau \in [-\pi, \pi]$, while for the translation the DOFs are given values $\delta \in [-0.5\text{ m}, 0.5\text{ m}]$.

An interesting case of unobservability occurs for planar movement of a sensor rig, when all the visible planes are perpendicular to the plane of movement. In such a case, it can be clearly seen that there is a free degree of freedom for the translation since the rank of the matrix concatenating the plane's normal vectors will always be deficient. This situation arises for a vehicle with an horizontal LRF which only observes vertical planes. In order to calibrate such a system, the scene should contain oblique planes, or the rig should be tilted in order to take observations from non-vertical planes.

Finally, the ratio $\eta = \mu_{6(m-1)}/\mu_1$ between the smallest and largest eigenvalues of the FIM is also an indicator of how well distributed the measurements are along the different directions (DoF) of the domain ($\eta = condition\_number^{-1}$). So, in the best case $\eta = 1$ which means that all plane observations are equally distributed in the space, while when $\eta \to 0$, the system becomes ill-conditioned.

2.3.1.5 Convergence

Considering that the calibration is observable, another important issue is to know if the solution converges to the correct value. This problem is not trivial for a non-linear optimization whose domain is not convex, containing local minima, as it is the case here. The local convexity of the error function around the solution depends on a number of parameters including: the configuration of the sensor rig, the amount of corner observations and their positions, and the noise in the sensor measurements. Thus, a mathematical condition for the convergence cannot be established in general.

However, given a configuration of the rig and a set of observations (COs), we can sample the optimization domain to conduct a qualitative analysis of its convexity. In figure 2.8 we display the residual error of eq. (2.18) for a rig with two LRFs which observe 20 *CO*s from different orientations. The error is shown with respect to the six parameters of the calibration by grouping pairs of DoFs in rotation and translation. The first row of figure 2.8 shows three sections of the sphere of possible rotations, corresponding to the planes $x-y$, $y-z$ and $z-x$, while the second row shows the translation domain, where each DoF takes values in [0.5, -0.5] meters around their true value at the center $\{0, 0\}$ of each graph. The resulting residuals are shown with a 2D heat map with the contour lines. In all the graphs, we see clearly a minimum at the correct calibration. We observe that there are local minima in the orientation domain, what implies that the initial values for the relative rotation must be given in a local region around the solution. On the other hand, the problem is convex for the translation as it is inferred from the formulation (the error depends linearly on the translation), therefore, the result does not depend on its initialization.

The test above has been repeated for a number of rig's configurations and for different *CO*s at different positions, and the results are qualitatively similar. For all such tests, we observe a similar trend for the error surfaces, indicating that there is at least one local minimum for the error function at the correct solution, and that there are local and global minima distributed in the domain. A particularly interesting case of wrong convergence occurs when the relative rotation of a pair of LRFs is initialized in a way that their scanning planes coincide. Note that there exists a global minimum for such set of parameters, where the error will be zero no matter the *CO*s. Apart from this point of degeneracy, initializing the calibration near this local or global minima can drive the optimization to an undesirable result, but in general, we observe that if the optimization starts from a point near the solution it will converge to the correct calibration. From these results, we can conclude that the convergence region is wide enough to be able to provide good initial values for the relative poses from simple visual inspection of the rig.

Another interesting point is to know if the calibration can be performed only from plane observations since, in fact, co-planarity constraints already restrict both the rotation and the translation of the relative poses in the rig. Several tests have indicated that this form of calibration is not possible because the cost function is not locally convex near the solution, and thus the problem cannot be properly constrained. As shown in figure 2.9, the introduction of the orthogonality constraint (eq. 2.12) makes the correct solution lie on a local minimum. Figure 2.9 shows the convergence error maps of different cost functions from: a) co-planarity constraints, b) orthogonality constraints, and c) a combination of both, which is actually the sum of the previous two. Note how these two functions complement well making the correct calibration lie on a local minimum.

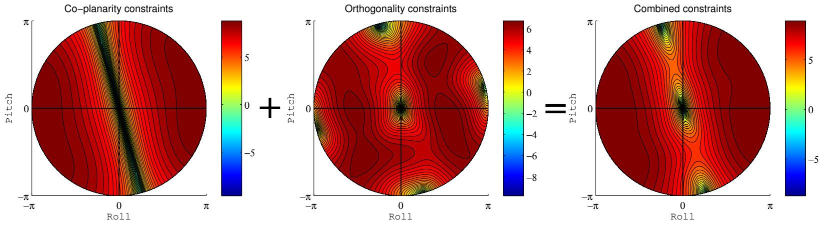


Figure 2.9: Error maps for different cost functions in logarithmic scale: a) co-planarity constraints, b) orthogonality constraints, c) the combination of these two. These graphs correspond to the previous simulation for two DoFs in the rotation domain (the graph *c* is the same as the graph shown at the top-left of figure 2.8). Note that the correct calibration at (0,0) lies on a local minimum only when the orthogonality constraints are applied.

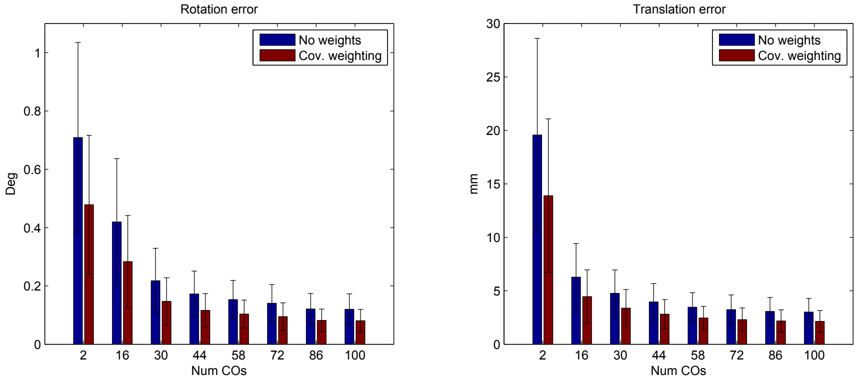


Figure 2.10: Calibration error of 2 LRFs with (red) and without (blue) covariance weighting, represented by the norms of the rotation (left) and translation (right) error vectors.

2.3.1.6 Experiments

A number of experiments has been carried out to validate the present approach from both simulated and real LRF rigs.

Simulation

In our simulation environment, a rig consisting of two non-parallel LRFs is placed at different distances and orientations with respect to a corner in order to gather measurements from several poses. The sensors are modelled according to the parameters of the Hokuyo UTM-30LX rangefinder, and the observations are generated with unbiased, uncorrelated Gaussian noise with $\sigma = 0.03$ m. The line features and their covariances are extracted from these synthetic observations. The calibration is esti-

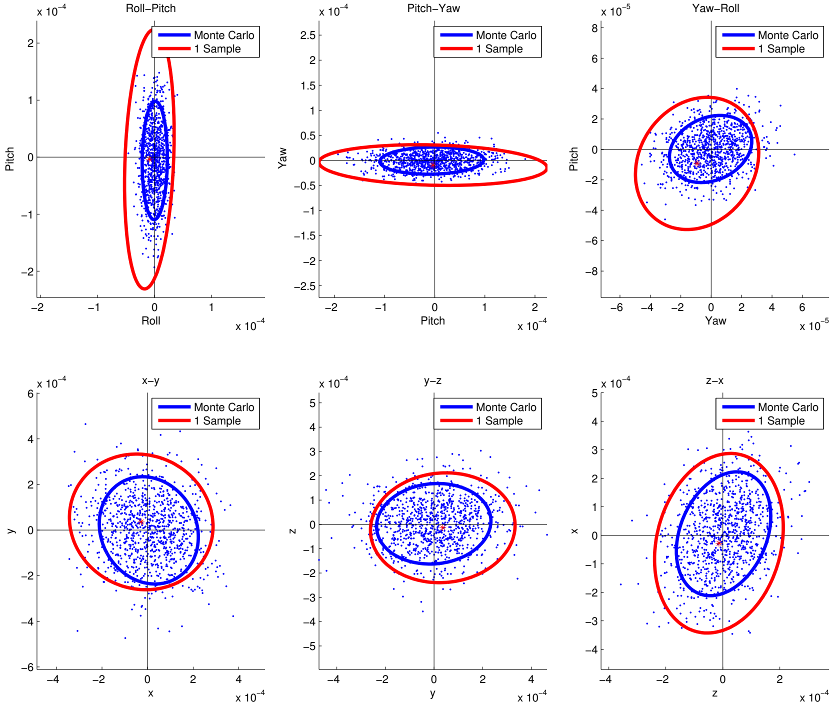


Figure 2.11: Error distribution and 2σ covariance ellipses of the calibration of two LRFs. A Monte Carlo simulation of 10^3 samples is shown (blue). One of this samples is also drawn (red cross) with the covariance computed by the calibration method.

mated for the cases of weighted (MLE) and unweighted optimization (standard least squares) for a varying number of *CO*s. The average errors of the calibration with respect to the true poses are obtained from a Monte Carlo simulation with 10^3 trials for every set of *CO*s. For each test, the initial relative pose is uniformly generated around the groundtruth at distance $d \in [0, 1\text{m}]$ and at an angle $|\tau| \in [0, \pi/4]$. The average errors of the relative rotation and translation are shown in figure 2.10 in degrees and millimeters, respectively. We observe that these errors diminish asymptotically with the number of *CO*s. Also, we see how the MLE solution that takes into account the covariance of the measurements is consistently more accurate than the solution which ignores that information. This test was repeated for several configurations of the LRF rig (different relative poses between the sensors) obtaining similar results.

We also study the bias and covariance of our method from the above Monte Carlo simulation by analysing the distribution of the calibration results. The six dimensional

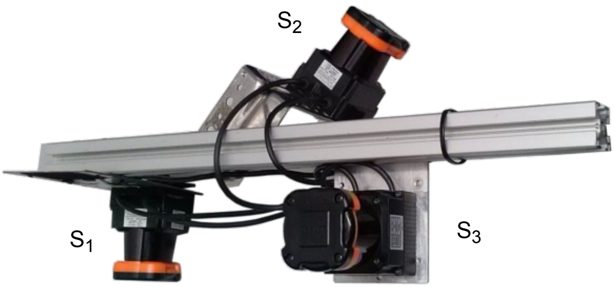


Figure 2.12: Test LRF rig with three Hokuyo UTM-30LX.

errors of the calibrated poses are shown in figure 2.11, by grouping pairs of DoF for the rotation and the translation, respectively. This figure shows the distribution of the 10^3 samples around the groundtruth (blue dots), and the 2σ confidence ellipses of Monte Carlo (blue ellipse), and the one corresponding to the estimated covariance of one sample (red ellipse) through the Cramér-Rao Bound (see appendix D). We can see how the bias of the method is very small with respect to the covariance.

Real data

We have also validated the proposed calibration method in real case scenarios employing: 1) a rig with three LRFs and 2) the sensors mounted on two autonomous cars. The characteristics of the calibrated LRFs are shown in table 2.1.

Table 2.1: Properties of the LRFs calibrated in this section.

	Hokuyo UTM-30LX	*Sick LMS 291-S05*
Range (m)	[0, 60]	[0, 80]
σ (m)	0.03	0.01
Resolution	0.25°	0.25°
Field of view	270°	180°

Test rig

In the first case, the test rig is composed of 3 Hokuyo UTM-30LX (see figure 2.12). The sensors' synchronization effect is neglected in this test since the rig is smoothly waved at a low velocity while the LRFs scan at a frame rate of 40 Hz (see the video at `http://youtu.be/YG4ShgyIUHQ`). The accuracy of the resulting calibration cannot be estimated directly since a groundtruth for the sensors relative poses is not available. Instead, we evaluate the accuracy of the method by checking that the pose composition from calibrating the different pairs closes a loop ($R_{12}R_{23}R_{31} = I$ and $\mathbf{t}_{12} + \mathbf{t}_{23} + \mathbf{t}_{31} = \vec{0}$).

In this test we validate our approach using a varying amount of *CO*s. The initial poses required by our method are given around a guess obtained from visual check of the rig, in a range of [0, 40] degrees for the rotation and [0, 1] meters for the translation, with respect to the correct calibration (several initializations are tested to check the robustness of our method). Table 2.2 shows the results of this test for different numbers of *CO*s, from a minimum of 2 *CO*s (that were extracted from a single observation, like the one represented in figure 2.7), to 100 *CO*s. The first three columns show the average residuals of the calibration of each pair of sensors, and the last two columns show the average deviation with respect to the loop closure condition of the three independent calibrations. From this table, we observe that, as expected, the residuals and the loop closure deviations decrease with the number of *CO*s.

Table 2.2: Residual errors for different calibrations for a varying amount of *CO*s. (*from one single observation).

*CO*s	res_{12}	res_{23}	res_{31}	R *dev (deg)*	**t** *dev (cm)*
2*	2.74	3.81	1.41	1.03	5.31
20	1.48	1.70	1.25	0.63	1.72
40	1.46	1.66	1.23	0.51	0.54
60	1.39	1.66	1.22	0.49	0.34
80	1.33	1.62	1.22	0.48	0.29
100	1.32	1.62	1.21	0.47	0.27

In this experiment, we have also calibrated the three LRFs by optimizing the full graph of constraints between them, so that the above loop closure condition is guaranteed. This way, the calibration should be more accurate since it uses all the information available. Table 2.3 shows the deviation of the relative pose between each pair of sensors and this global calibration. The deviations between the relative poses are expressed in degrees for the rotations (r_{12}, r_{23} and r_{31}) and in centimetres for the translations (t_{12}, t_{23} and t_{31}).

The covariance of the resulting calibration depends on the information provided by the *CO*s. In general, providing more *CO*s contributes to reduce the uncertainty of the solution. This is confirmed in figure 2.13, which displays the maximum eigenvalue of the calibration covariance with respect to the number of *CO*s for the experiment above. We observe how the value of the variance decreases asymptotically with the number of *CO*s. This feature is relevant since it allows the user to set the maximum uncertainty for the calibration, so that the process of gathering *CO*s stops after such a limit is reached.

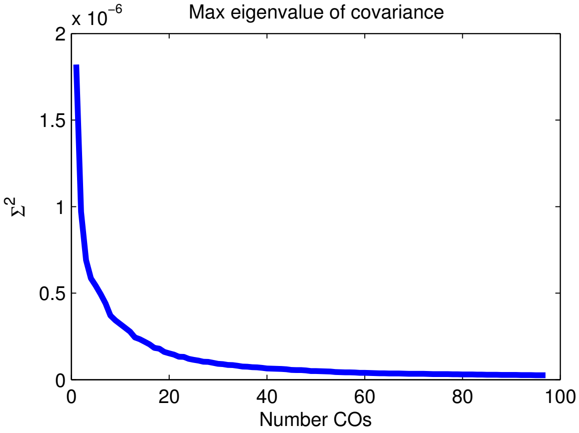


Figure 2.13: Maximum eigenvalue of the calibration covariance with respect to the number of *CO*s.

Table 2.3: Deviations between global calibration and the calibration of each pair for a varying amount of *CO*s (*from one single observation).

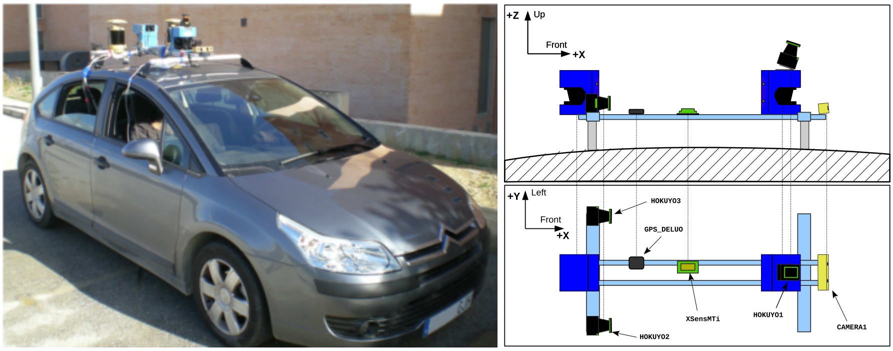
*CO*s	r_{12}(deg)	t_{12}(cm)	r_{23}(deg)	t_{23}(cm)	r_{31}(deg)	t_{31}(cm)
2*	0.84	1.21	0.54	1.03	0.65	0.83
20	0.41	1.10	0.40	0.93	0.52	0.72
40	0.40	0.96	0.36	0.91	0.50	0.74
60	0.39	0.89	0.35	0.79	0.43	0.65
80	0.33	0.88	0.33	0.68	0.29	0.66
100	0.32	0.73	0.34	0.67	0.27	0.61

Autonomous car datasets

We have also validated our method by calibrating the sensors mounted on two different autonomous vehicles, using two publicly available datasets[7,8]. For the dataset in [Blanco-Claraco *et al.*, 2014], the vehicle has five LRFs in total, three Hokuyo UTM-30LX and two Sick LMS 291-S05, whose configuration is shown in figure 2.14. The Sick sensors scan horizontal planes, and therefore, the calibration cannot be fully constrained unless they observe non-vertical planes (for that, either the rig must be tilted or the scene should contain oblique planes like shop awnings). This situation does not occur in the dataset, so, only the Hokuyo sensors are considered. Note that two of these three sensors (labelled as Hokuyo2 and Hokuyo3) scan almost the same vertical plane, however they can still be calibrated since the sensor Hokuyo1

[7] http://www.mrpt.org/MalagaUrbanDataset

[8] http://grandchallenge.mit.edu/wiki/index.php?title=PublicData



(a) (b)

Figure 2.14: A semi-autonomous car which incorporates several laser rangefinders at different orientations. b) Scheme of the sensors.

has a different orientation. For calibration, we have chosen an extract of the dataset[9] where the car travels through some streets with buildings at the sides. The corner observations where selected in a supervised way because the clutter in the scene (other cars, trees, etc.) introduces a huge amount of wrong correspondences that prevents a correct corner detection. Our method could be applied however automatically when there is less clutter, like in the previous experiment.

The calibration was computed from 12 *CO*s taking the extrinsic parameters provided with the dataset for the initialization which, according to the authors, were manually measured from the rig. All the corners selected for this calibration come from the floor and a wall, which are assumed to be perpendicular. The average angle between these planes for the 12 corners after calibration was 89.4 degrees, with a standard deviation of 0.68, while, by using the calibration provided with the dataset the average angle was 85.9 degrees, with a standard deviation of 5.6. Also, the visualization of the calibrated laser scans for both cases shows that our estimation is clearly more accurate since the intersection of the scanning planes produces coincident points, and the straight segments observed by each laser lie on the scene planes. On the contrary, we observe that the alignment is not so good for the calibration proposed in the dataset.

The second dataset used here [Huang *et al.*, 2010] corresponds to the recordings of a car which participated in the Darpa Challenge, which has 13 LRFs. The dataset was taken in open outdoor spaces which are scarce in corner structures. At some point of the video sequence however, the car passes near a building where a pair of LRFs observe two corners (see figure 2.15). Such corners are seen at the time 15m03s of the dataset “2007-11-03-log-uce-scrubbed.mission1”, by the lasers 2 and 4. From

[9] http://www.youtube.com/watch?v=qZMlc5UeUpE

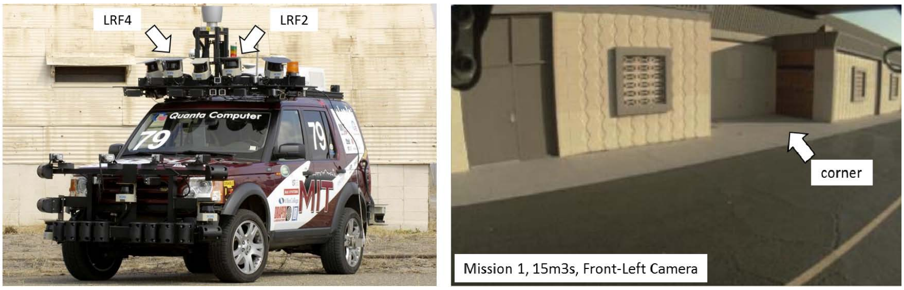


Figure 2.15: MIT's autonomous vehicle Talos (left), and a snapshot of the corner observed by the lasers 2 and 4 of this vehicle during the 2007 Darpa Urban Challenge (right).

this single observation we perform calibration using different initializations for their relative pose. Such initializations were randomly generated around the calibration proposed in the dataset, in a similar way to the previous experiments. In this case, the lines were segmented using a parameter independent line fitting method [Prasad *et al.*, 2011], which works better given the low angular resolutions of the sensors in the dataset (1/4 of the maximum resolution). The estimated pose differs from the one reported by the authors by $\sim 0.8^\circ$ in the rotation and ~ 20 cm in the translation. It is hard to say which calibration is more accurate in terms of the rotation, while for the case of the translation, the calibration provided with the dataset looks more accurate according to the visualization of the reconstructed laser scans. We attribute this difference to the *CO*s obtained (only 2) and to the curvature of one of the observed "planes" (the floor). Also, the observations are not simultaneous since the scans were taken with no synchronization while the car moves at a considerable speed, contributing to increase the calibration error.

There are several ways to improve the calibration obtained for this practical example. The first thing would be to take observations in a more structured scenario, like in a city, or just in front of a wall (a wall with the floor constitute a corner). In this way, a bigger number of observations could be taken to compensate for different sources of error. Finally, taking synchronized (or still) observations will also help to obtain more accurate results.

Table 2.4: MIT Dataset calibration from a single observation with different initializations.

Av Rot deviation	0.79°
Rot precision	0.02°
Av Trans deviation	23.2 cm
Trans precision	3.65 cm

2.3.1.7 Discussion

We have presented, to the best of our knowledge, the first general solution to calibrate the extrinsic parameters of a rig of 2D range scanners. The method relies on the observation of perpendicular planes to constraint the relative poses of the different LRFs. This problem is solved in a probabilistic framework that takes into account the uncertainty in the measurements of the sensors, and as a result, it also provides the uncertainty of the estimated calibration. The observability and the convergence conditions for the problem are studied, showing that there exists a minimal solution which only requires a single observation from the LRF rig.

The calibration method proposed here presents important advantages with respect to previous approaches, since it is applicable to almost any sensor configuration, it is easy to use and easy to automatize, while being robust and accurate. Also, its probabilistic formulation allows to calibrate different models of sensors, as each error is weighted according to its uncertainty. We have conducted several experiments to validate our approach, both with synthetic and real data, which have demonstrated the claimed features of our proposal.

2.3.2 Calibration of a set of 3D range cameras

The integration of several 3D range cameras (or RGB-D cameras) in a mobile platform is useful for applications in robotics and autonomous vehicles that require a large field of view. This situation is increasingly interesting with the arrival of low cost range cameras like those developed by Primesense. In this context, the available methods for extrinsic calibration present mainly two types of disadvantages: they have restrictions on the camera positioning (e.g. requirement of overlapping), or they rely on the tracking of the camera trajectory, which can be tedious to obtain, besides having issues of robustness and accuracy. The disadvantages of previous calibration approaches were clear after the construction of a device for omnidirectional intensity and range image acquisition based on a rig of RGB-D cameras (figure 5.1) [Fernández-Moral *et al.*, 2014b; Gokhool *et al.*, 2014]. This new sensor, which is used in chapter 5 for simultaneous localization and mapping, prompted in us the need of a robust and easy calibration method, since the accuracy of the parameters from the construction design were not satisfactory, and the solutions proposed in the literature were not suitable for our problem due to the limitations commented above.

In this section we propose a new uncomplicated technique for extrinsic calibration of range cameras that relies on finding and matching planes. The method that we present serves to calibrate two or more range cameras in an arbitrary configuration (overlapping is not needed), requiring only to observe one plane from different viewpoints. The conditions to solve the problem are studied, and several practical examples are presented covering different geometric configurations, including an omnidirectional RGB-D sensor composed of 8 range cameras. The quality of this calibration is evaluated with several experiments achieving successful calibrations. Such experiments demonstrate that our method constitutes a versatile solution that is extremely fast and easy to apply.

2.3.2.1 Related works

To put our work into context, we review first some relevant approaches to this problem. A classical strategy for extrinsic camera calibration is through the detection and matching of control points that are detected in the overlapping regions of the different cameras [Szeliski and Shum, 1997]. However, the overlap requirement constitutes a very strong constraint, specially the goal is to enlarge the field of view. Besides, even when some overlap exists, it is generally more complicated to match features in range images than in intensity images. A different strategy that have been widely used for different calibration problems consists of using a calibration pattern to infer constraints on the sensor relative poses. Such a procedure has also been applied to RGB-D sensors [Basso *et al.*, 2014a; Macknojia *et al.*, 2013] with the same above limitation for overlapping field of views. Besides, the need of calibration pattern itself is wearisome.

A more general approach which does not depend on the camera set-up is based on ego-motion to match the camera trajectories, which are tracked independently, [Brookshire and Teller, 2012; Heng *et al.*, 2013; Schneider *et al.*, 2013]. Such approaches rely on the SLAM or visual odometry (VO) robustness, which depends highly on the environment, especially for range-only cameras. Besides, obtaining a useful trajectory is far more tedious and inconvenient than taking a few images from different positions as the technique we propose here.

Contribution

We present a new method for extrinsic calibration of range cameras that avoids the problems mentioned above. Our method relies on matching planes that can be observed simultaneously from different cameras. For that, only one plane has to be observed from different camera locations. This approach has several advantages as the calibration can be performed very quickly and robustly, it does not require any calibration pattern but a single plane from the environment (the floor, the ceiling, a wall, ...), and supervision is not required. In this work, we test the performance of the method by calibrating two typical configurations of range cameras, demonstrating very satisfactory results in all the cases.

In the following we give the details of our calibration approach, the segmentation and parametrization of the planes and their matching. The observability of the problem is studied next. Then, the equations for extrinsic calibration are derived for a pair of cameras (section 2.3.2.3) and for an arbitrary number of cameras (section 2.3.2.6). For both cases, calibration results for different camera configurations are presented. Finally, the conclusions are outlined.

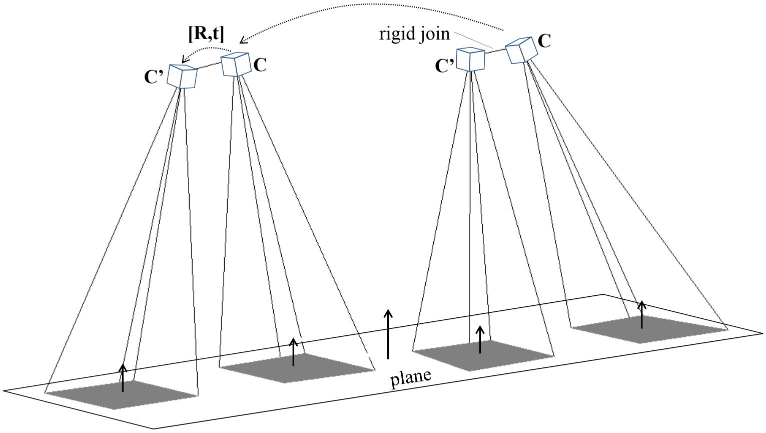


Figure 2.16: A planar surface is observed by two range cameras rigidly joined from different positions. In this way, plane correspondences are captured from different orientations with respect to the rig's reference system to perform extrinsic calibration.

2.3.2.2 Calibration approach

We propose to solve the extrinsic calibration between several "a-priori" non overlapping range cameras by matching planar features that are seen from different viewpoints. We take advantage of the fact that structured environments contain large planes (e.g. the floor, walls, ceiling) that can be reliably observed by the different sensors simultaneously, making use of such planes to establish correspondences (see figure 2.16). With this strategy we avoid the need of creating a specific calibration pattern for the sensor set-up. Also, no SLAM neither odometry are needed, avoiding robustness issues and making the procedure much more accessible and easy to use.

Before dealing with the extrinsic calibration itself, related issues like the plane segmentation, parametrization and matching are addressed next.

Plane segmentation and parametrization

In order to obtain planes (planar patches to be precise), the depth images are segmented with a region growing approach [Holz and Behnke, 2013]. This technique is used here due to its efficiency to segment organized images, however other methods for plane segmentation can be used equally [Zuliani *et al.*, 2005; Borrmann *et al.*, 2011].

A planar patch is represented by its normal vector $\mathbf{n}$, with $||\mathbf{n}|| = 1$, and the distance d to the optical center of the camera. In this way, a point $\mathbf{p}$ lying on the plane fulfils the equation

$$\mathbf{n} \cdot \mathbf{p} + d = 0 \tag{2.28}$$

This overparametrization is very convenient for the formulation of the calibration errors in the next sections.

The plane parameters and their covariances are estimated following [Poppinga *et al.*, 2008], assuming accurate directions of measurements $\mathbf{m}_i$, where the noise only affects the range measurements ρ_i. After the intrinsic calibration has been performed, we can assume that $\rho_i \sim N(\hat{\rho}_i, \sigma_i)$, where $\hat{\rho}_i = d/\mathbf{n} \cdot \mathbf{m}_i$ is the true range of the i-th measurement. The standard deviation σ_i is generally a function that depends on the range ρ_i and on the incidence angle $\sigma(\rho_i, \mathbf{n} \cdot \mathbf{m}_i)$. However, in this work we make the same simplification as in [Poppinga *et al.*, 2008] to assume the standard deviation σ_i independent on $\{\mathbf{n}, d\}$, and estimate σ_i in a conservative way: $\sigma(\rho_i, \mathbf{n} \cdot \mathbf{m}_i) < \sigma$. From this simplification the plane parameters and their covariances can be analytically defined. Thus, the optimal $\mathbf{n}^*$ is the eigenvector corresponding to the smallest eigenvalue of the matrix

$$M = \sum_{i=1}^{N} (r_i - r_G)(r_i - r_G)^\top \tag{2.29}$$

where

$$r_G = \frac{1}{N} \sum_{i=1}^{N} r_i \tag{2.30}$$

is the gravity center of the plane pixels. The optimal d^* is given by

$$d^* = \mathbf{n}^* r_G \tag{2.31}$$

and the covariance of the plane parameters $\Sigma^* = (H)^+$ is calculated as the Moore-Penrose generalized inverse of

$$H = \frac{1}{\sigma^2} \sum_{i=1}^{N} \begin{bmatrix} r_i r_i^\top & -r_i \\ -r_i^\top & 1 \end{bmatrix} \tag{2.32}$$

The simplification of considering constant variance (i.i.d.) assumed above can be substituted for a more realistic model [Pathak *et al.*, 2010c] to obtain more accurate results. But this requires a complex numeric calculation of the plane parameters and their covariances, which is out of the scope of this section.

Constraint equations

The constraint equations used to calibrate the sensors are inferred from plane correspondences. A plane correspondence is defined here as the simultaneous observation of a planar surface by at least two range cameras. To illustrate this, consider a pair of rigidly jointed range cameras C and C', where the camera C represents the system of reference, and C' is located with a relative transformation $[\mathrm{R}|\mathbf{t}] \in \mathbb{SE}(3)$ with respect to C, where the rotation $\mathrm{R} \in \mathbb{SO}(3)$ is represented with a 3×3 matrix and the translation $\mathbf{t} \in \mathbb{R}^3$.

Orientation constraint: A constraint for the relative orientation between the two sensors is stated from the observed normal vectors

$$\mathbf{n} - \mathbf{R}\mathbf{n}' = \vec{0} \tag{2.33}$$

being $\mathbf{n}$ and $\mathbf{n}'$ the observed normal vectors seen by the cameras C and C' respectively.

Position constraint: A constraint for the relative position is given by

$$d - d' + \mathbf{n} \cdot \mathbf{t} = 0 \tag{2.34}$$

where d and d' are the observed distances from the plane to the optical centers of the depth cameras C and C'.

Obtaining plane correspondences

Similarly as in the previous calibration problem (section 2.3.1.2), the sensor relative poses must be known in order to establish the plane correspondences. Thus, the problem consists of estimating the calibration and plane correspondences simultaneously. For that, all the plane observations gathered from a single observation of the rig are matched between them, so that they will contain both correct and wrong correspondences. Then RANSAC [Fischler and Bolles, 1981] is applied to find the extrinsic calibration with a larger number of supporting correspondences, discarding the rest of correspondences as outliers. This procedure is carried out in two steps: first, the outliers showing a large error in the orientation are discarded, and second, those outliers in distance are removed (this order is chosen since the noise in the orientation of the normal vectors is typically smaller than that in the plane position). For these RANSAC processes, the relative poses between the pair of cameras are calculated from a sample of 3 non-degenerate plane correspondences using the models defined in section 2.3.2.3.

Note that giving an initial estimation for the relative position of the cameras can also be applied to facilitate the matching of plane observations. Also, there exist other plane matching strategies can be applied avoiding the need of an initial estimate for the calibration [Pathak *et al.*, 2010b; Fernández-Moral *et al.*, 2013b].

The process for gathering correspondences is performed automatically while the camera rig is moving until the problem is well conditioned according to the Fisher Information Matrix, as explained in section 2.3.2.4. The range cameras synchronization effect is neglected in this work since the images are captured at a minimum frame rate of 30 Hz, and the camera rigs are never moved abruptly.

2.3.2.3 Problem formulation

Given a set of plane correspondences gathered from two rigidly jointed range cameras C and C', as defined above, and provided that the correspondences fulfill the observability condition (concretely the one represented by eq. 2.49), we want to estimate the optimal $[\mathrm{R}|\mathbf{t}]$ assuming that the measurements are affected by unbiased Gaussian noise as modelled in 2.3.2.2. This problem can be divided into two separate ones since the rotation and the translation restrictions are decoupled.

Solving for the rotation

The maximum likelihood estimation (MLE) of the relative rotation R is given by the maximization of the log-likelihood

$$\underset{\mathrm{R}}{\operatorname{argmax}} \left(\ln \prod_{i=1}^{N} p(\mathbf{n}_i, \mathbf{n}'_i | \mathrm{R}) \right) \tag{2.35}$$

for N plane correspondences, where the likelihood of the rotation for the i-th correspondence is expressed as

$$p(\mathbf{n}_i, \mathbf{n}'_i | \mathrm{R}) = \frac{1}{\sqrt{(2\pi)^3 |\Sigma_i|}} \exp\left(-\frac{1}{2} (\mathbf{n}_i - \mathrm{R}\mathbf{n}'_i)^\top \Sigma_i^{-1} (\mathbf{n}_i - \mathrm{R}\mathbf{n}'_i) \right) \tag{2.36}$$

being $\mathbf{n}_i$ and $\mathbf{n}'_i$ the observed normal vectors from the plane i as seen by the cameras C and C' respectively; R is the rotation matrix in $\mathbb{SO}(3)$, and Σ_i is the 3×3 covariance block corresponding to the normal vector of the plane correspondence (calculated from the fusion of both observations [Pathak *et al.*, 2010c], see appendix C). Considering independent errors of the plane correspondences, the derivation of this MLE coincides with the solution of the least squares problem expressed as

$$\underset{\mathrm{R}}{\operatorname{argmin}} \sum_{i=1}^{N} \omega_i \|\mathbf{n}_i - \mathrm{R}\mathbf{n}'_i\|^2 \tag{2.37}$$

where ω_i is the weight of the plane correspondence

$$\omega_i = \frac{1}{|\Sigma_i|} \tag{2.38}$$

This problem is similar to the one of estimating the rotation of a registered set of 3D points [Arun *et al.*, 1987]. Thus, employing the same procedure, the above equation can be expressed as

$$\begin{aligned} \mathrm{R} &= \underset{\mathrm{R}}{\operatorname{argmin}} \left(\sum_{i=1}^{N} \omega_i \mathbf{n}_i^\top \mathbf{n}_i - 2 \sum_{i=1}^{N} \omega_i \mathbf{n}_i^\top \mathrm{R}\mathbf{n}'_i + \sum_{i=1}^{N} \omega_i \mathbf{n}_i'^\top \mathbf{n}'_i \right) \\ &= \underset{\mathrm{R}}{\operatorname{argmin}} \left(-2 \sum_{i=1}^{N} \omega_i \mathbf{n}_i^\top \mathrm{R}\mathbf{n}'_i \right) \\ &= \underset{\mathrm{R}}{\operatorname{argmax}} \sum_{i=1}^{N} \omega_i \mathbf{n}_i^\top \mathrm{R}\mathbf{n}'_i \end{aligned} \tag{2.39}$$

that can be denoted as

$$\sum_{i=1}^{N} \omega_i \mathbf{n}_i'^{\top} \mathrm{R}\mathbf{n}_i = trace(WY^{\top}RX) \tag{2.40}$$

where $W = diag(\omega_1, ..., \omega_n)$ is an $n \times n$ diagonal matrix containing the weights ω_i; and Y and X are $3 \times n$ matrices with the normal vectors $\mathbf{n}_i'$ and $\mathbf{n}_i$ as their columns, respectively. This problem is solved with singular value decomposition (SVD) over the 3×3 covariance matrix

$$S = XWY^{\top} \tag{2.41}$$

From the singular value decomposition $S = U\Sigma V^{\top}$, the rotation is obtained as

$$\mathrm{R} = V \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det(VU^{\top}) \end{pmatrix}}_{A} U^{\top} \tag{2.42}$$

where the matrix A is used to convert the degenerate case of a reflection

$$det(VU^{\top}) = -1 \tag{2.43}$$

into a valid rotation in $\mathbb{SO}(3)$. For further details on the mathematics, please refer to [Arun *et al.*, 1987; Sorkine, 2009].

Solving for the translation

The MLE of the translation is obtained by maximizing the log-likelihood associated to the probability

$$p(\mathbf{n}_i, \mathbf{n}_i', d_i, d_i' | \mathbf{t}) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{1}{2} \frac{(d_i - d_i' + \mathbf{n}_i \cdot \mathbf{t})^2}{\sigma_i^2} \right) \tag{2.44}$$

where d_i and d_i' are the observed distances from the plane i to the optical centers of the depth cameras C and C' respectively, σ_i^2 is the error variance, and $\mathbf{t}$ is the relative translation we are looking for. This is equivalent to the least squares problem

$$\operatorname*{argmin}_{t} \sum_{i=1}^{N} \omega_i (d_i - d_i' + \mathbf{t} \cdot \mathbf{n}_i)^2 \tag{2.45}$$

with the weight given by $\omega_i = 1/\sigma_i^2$. This has a closed form solution given by

$$\mathbf{t} = -H^{-1} g \tag{2.46}$$

where H and g are the Hessian and the Gradient of the error function respectively, which are calculated as

$$H = \sum_{i=1}^{N} J_i^{\top} W_i J_i \ , \ \ g = \sum_{i=1}^{N} J_i^{\top} W_i \mathbf{r}_i \tag{2.47}$$

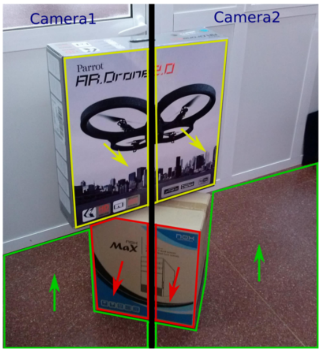


Figure 2.17: A particular set-up from which we can calibrate the cameras with a single observation. The planar patches on the left and right are those extracted from the two cameras.

where the Jacobians, the weights and the residuals are calculated from

$$J_i = n_i^\top \ , \ \ W_i = \frac{1}{\sigma_i^2} \ , \ \ \mathbf{r}_i = d_i - d_i' \tag{2.48}$$

2.3.2.4 Observability

It can be seen that each plane correspondence imposes three new constraints between the pair of sensors: two for the relative rotation and one for the relative translation. Thus, we need at least 3 measurements from linearly independent plane observations (i.e the observed normal vectors of the planes must be linearly independent) to compute the relative pose of a pair of sensors (only two measurements are needed to compute the rotation), and a minimum of $3(N-1)$ correspondences to calibrate a rig with N sensors. To put a simple example, let's consider a single sensor observing the corner of a room. The observation of the three perpendicular planes gives us enough information to localize the camera and its relative motion with respect to a previous pose. Analogously, the relative pose between two cameras can be obtained if they observe 3 plane correspondences with linearly independent normal vectors (either observed from one view, like in figure 2.17, or from several ones). The most simple and convenient procedure of calibration would be to take a short sequence of images of one big plane at different orientations of the rig (see the video at `http://youtu.be/MGydi5R7ldA`).

Similarly as for the calibration of 2D LRFs, here we make use of the Fisher Information Matrix (FIM) to identify those unobservable cases in which the calibration cannot be determined. This analysis defines how the measurements should be taken to avoid these situations. As we will see, the probability of the MLE is given by an unbiased Gaussian distribution (this assumption is realistic only after intrinsic correction). For this estimator (called efficient [Fernández-Madrigal and Claraco, 2013]), the FIM coincides with the Hessian of the least squares problem resulting from the MLE, and

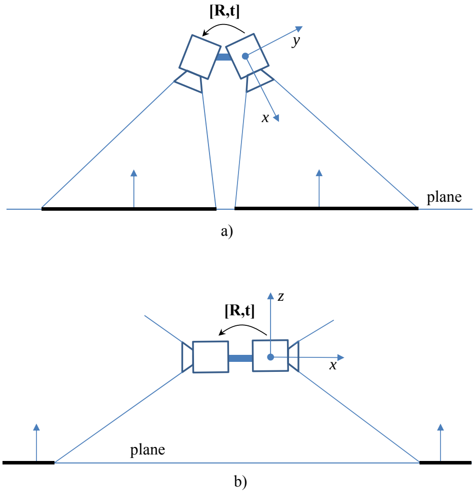


Figure 2.18: Different sensor configurations with a pair of cameras: a) Adjacent cameras, b) Opposite cameras.

its inverse is the covariance of the resulting calibration (see the appendix D). When the FIM is singular, the information provided is not sufficient and the MLE does not exist. For a pair of cameras, it can be verified that when the FIM is not singular, then

$$rank(\sum_{i=1}^{N} \mathbf{n}_i \mathbf{n}_i^\top) = 3 \tag{2.49}$$

where n_i is the normal vector of the plane i as seen from one of the cameras in the pair.

From our experiments, we have verified that the covariance of both the rotation and the translation estimations decrease asymptotically as the number of plane correspondences increases. The covariance is used as the condition to control the calibration convergence, and hence, to stop gathering plane correspondences. In our tests, we stop this calibration when the maximum eigenvalue of the covariance is under 10^{-3}, which has shown to be a good compromise between accuracy and effort to obtain plane correspondences.

2.3.2.5 Practical study cases

1. Adjacent cameras

This case is interesting to provide a larger field of view of the scene, being specially practical for low cost sensors like Asus Xtion (see figure 2.18.a). This case serves us to illustrate the conditioning of the problem, and so to show different possibilities for calibration. One of this situations is the calibration of the pair from one single observation, i.e. without moving the rig. This is only possible if three planar patches whose normal vectors span through the different directions of the space are visible at the same time by both cameras. This case can be easily set-up, as the example shown in figure 2.17.

In practice, however, it is even more convenient to take several images from different orientations pointing to one single plane (the floor, for example), since we can gather more quickly enough plane correspondences that help to reduce the error from the measurement noise. This may take no longer than 2 or 3 seconds.

In table 2.5 we show an example of how the average residual error is reduced when raising the number of plane correspondences. The alignment errors in rotation and translation are measured in a dataset containing 2K correct plane correspondences for the five calibrations, the plane correspondences were taken in all directions of the space.

Table 2.5: Residual errors for different calibrations using a different number plane correspondences.

Correspondences	*Av rot error (deg)*	*Av trans error (cm)*
3	1.12	1.89
10	0.68	1.01
30	0.52	0.82
60	0.49	0.74
100	0.49	0.61

2. Cameras in opposite directions

This case is interesting, for instance, for vehicles that need to observe the scene forward and backward. We address this case here also since it probably represents the most challenging case to obtain plane correspondences in different directions (notice that the further the viewing directions of the cameras are, the more difficult is to find plane correspondences). Figure 2.18.b shows how the plane correspondences can still be obtained to add constraints in the different directions of the space, for example, by rotating the camera rig. Calibration was performed automatically while the user waved the camera near the floor. After 5 seconds from the start of the experiment, the calibration finishes with 29 plane correspondences (see the video at `http://youtu.be/MGydi5R71dA`). In this case the deviation with respect to the rig parameters is less than 1 deg for the rotation, and in the order of millimeters for the translation.

3. Sensors of different types

Though most of our experiments are carried out with structured light Primesense cameras, other range sensors can also be calibrated with our method. Concretely, a time-of-flight camera and a Kinect sensor mounted on a robot are calibrated by moving the robot (figure 2.19, left) around to gather plane correspondences. The errors in the plane observations from both sensors will follow different distributions, so that they are weighted accordingly as said in section 2.3.2.2.

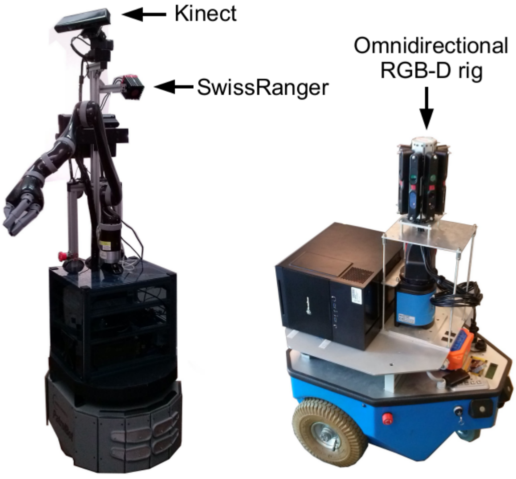


Figure 2.19: Robots which mount rigs of range and RGB-D cameras.

2.3.2.6 Extrinsic calibration of an arbitrary number of range cameras

This section extends the previous formulation for an arbitrary number M of range cameras. Note that for the case when there are no loop closures between the sensors, i.e. there is only one possible way to correlate the relative pose of any pair of sensors. The extrinsic calibration can be calculated as in the previous section by estimating the relative pose between each pair of adjacent sensors, and performing pose composition to place them in a common reference. Instead, this section is dedicated to the case in which there are plane correspondences that create loop closures between sensors. For the sake of space, we present directly the least squares equations, which as in the previous section, derive from the ML estimation.

The relative rotation between the different sensors can be formulated as

$$\underset{\{\mathrm{R},\mathbf{t}\}}{argmin} \sum_{j=1}^{M} \sum_{k=j+1}^{M} \sum_{i=1}^{N} \lambda_i(j,k) \Big((\mathrm{R}_j \mathbf{n}_i^j - \mathrm{R}_k \mathbf{n}_i^k)^\top \Sigma_i^{-1} (\mathrm{R}_j \mathbf{n}_i^j - \mathrm{R}_k \mathbf{n}_i^k) + \frac{1}{\sigma_i^2} (d_i^j - d_i^k - \mathbf{t}_j \mathrm{R}_j \mathbf{n}_i^j + \mathbf{t}_k \mathrm{R}_k \mathbf{n}_i^k)^2 \Big) \tag{2.50}$$

where j and k are indices of the M sensors and i is the index of each one of the N planes observed; $\lambda_i(j,k)$ is a binary variable that equals 1 when the plane i is observed by sensors j and k, being 0 otherwise; $\mathbf{n}_i^j$ and $\mathbf{n}_i^k$ are the normal vectors, and d_i^j and d_i^k are the distances of the camera optical center to the plane i observed from sensors j and k, respectively; Σ_i and σ_i are the covariance and the variance of the probabilities in eq. 2.36 and 2.44, respectively; and the relative poses between the sensors are represented by $\{\mathrm{R},\mathbf{t}\} \in \mathbb{SE}(3)$.

This least squares system has a different structure from the one in the previous section, that can not be solved with the strategy used from equations 2.39 to 2.42. Instead, we rewrite the problem to represent the relative rotations in minimal parametrization with the exponential map from Lie algebra (see appendix B), similarly as it was done for the calibration of 2D range scanners.

This is a non-linear least squares system that is solved iteratively with Levenberg-Marquardt as in equations 2.20-2.21, where the Jacobian and the vector of residuals are given by

$$\begin{aligned} J_i &= [0 \ldots 0\, J_i^{(j)}\, 0 \ldots 0\, J_i^{(k)}\, 0 \ldots 0] \\ J_i^{(j)} &= skew(\mathbf{n}_i^j)\ ,\ J_i^{(k)} = skew(-\mathbf{n}_i^k) \end{aligned} \tag{2.51}$$

thus, the Hessian H and the gradient $\mathbf{g}$ are calculated incrementally as

$$H = \sum_{i=1}^{N} H_i\ ,\ \ \mathbf{g} = \sum_{i=1}^{N} \mathbf{g}_i \tag{2.52}$$

which have the form

$$H_i = \begin{bmatrix} J_i^{jT}\omega_i J_i^j & & J_i^{kT}\omega_i J_i^j \\ & \ddots & \\ J_i^{jT}\omega_i J_i^k & & J_i^{kT}\omega_i J_i^k \end{bmatrix}\ ,\ \mathbf{g}_i = \begin{bmatrix} J_i^{jT}\omega_i r_i \\ \vdots \\ J_i^{kT}\omega_i r_i \end{bmatrix} \tag{2.53}$$

2.3.2.7 Calibration of a rig for omnidirectional image acquisition

We have designed a camera rig for omnidirectional RGB-D acquisition which comprises 8 Asus Xtion Pro Live (Asus XPL) sensors mounted in a radial configuration (see figure 2.19.b). This device motivated at the origin the work described in this section, since the parameters from the construction design were not accurate for our application. Existing calibration approaches like those based on SLAM [Brookshire and Teller, 2012] are very time consuming and impose important restrictions on the trajectory, since planar movement (as we have in our robot) is a degenerate case where calibration cannot be achieved. Thus, we employed the calibration method described in the previous section, which was applied on a sequence of images taken with the robot (planar movement is not a degenerate case in our approach).

The relative positions between the RGB cameras is the same as these between their corresponding depth cameras for our sensor configuration. Therefore, both RGB and depth omnidirectional images can be built as it is illustrated in chapter 5 (see figure 5.4). The 3D point cloud can be also built from such images as it is shown in figure 5.6(a).

The precision of calibration is tested with an experiment where the robot moves in a small circular trajectory ($\varnothing \sim 0.5$ m) in the center of a room, taking 200 images.

In table 2.6, the average residual in orientation and translation for the plane correspondences of these images is presented for different extrinsic calibrations: design parameters (no extrinsic calibration) with and without intrinsic correction, and the extrinsic calibration also with and without intrinsic correction. The residual of Iterative Closest Point (ICP) alignment of the spherical point clouds from these images is also shown. For all the cases, the combination of intrinsic and extrinsic calibration offers the best results.

Table 2.6: Residual errors for different combinations of intrinsic and extrinsic calibrations.

Calib / Error type	*Res. rot (deg)*	*Res. trans (cm)*	*Res ICP (cm)*
Design Specs	3.17	3.0	0.49
Design S.+Intrinsic	2.95	3.1	0.45
Extrinsic calib	1.78	2.9	0.34
Extrinsic+Intrinsic	1.60	2.5	0.29

2.3.2.8 Discussion

A new methodology for calibrating the extrinsic parameters of range camera rigs has been presented in this section. The method relies on the matching of plane observations from the different sensors. No constraints are put on the position of the cameras, where the only requirement for the system is that there is a planar surface that can be observed simultaneously. The observability conditions are analyzed, and a solution is presented based on MLE. With our method, performing calibration becomes very fast and easy for the user, avoiding problems of previous solutions which rely either on calibration patterns or trajectory estimation methods. The method has been tested for different configurations of cameras, including a camera rig designed for omnidirectional image acquisition. All the experiments have validated the claimed features of our proposal.

2.3.3 Calibrating a 3D range camera and a 2D laser scanner

After the previous solutions for calibrating sets of 2D and 3D range scanners, we present in this section a new method to calibrate a combination of these ones. The same strategy of the previous sections is applied to make use of plane observations to constrain the relative poses of the sensors. We also follow a similar procedure as before to formulate the problem and present an analysis on the restrictions and the observability conditions. Still, a full description of the approach is given since the problem characteristics are different from the previous ones. Experimental results are provided with both simulation and real mobile platforms with several range sensors.

2.3.3.1 Related works

There are some methods in the literature that have addressed the problem of extrinsic calibration between a laser and different sensors like RGB cameras, wheel odometry, etc. One that has received considerable attention is that of finding the relative pose between a laser and a RGB camera. The first solution to this problem was reported in [Zhang and Pless, 2004]. This solution employed a checkerboard as a calibration pattern, and estimates the extrinsic calibration by restricting that points observed with the laser lie on the same 3D plane where the checkerboard lies. When these assumptions hold, the relative position and orientation between both sensors can be estimated through these geometric restrictions. Some improvements of this former solution have been presented along the last decade by exploring different calibration patterns [Li *et al.*, 2007; Ha, 2012; Moreno *et al.*, 2013], decoupling rotation from translation [Zhou and Deng, 2012], presenting a minimal closed form solution [Vasconcelos *et al.*, 2012], and adopting different optimization strategies [Zhou, 2014].

Recently, a solution was presented to the problem of extrinsic calibration between a laser and a Kinect [Devaux *et al.*, 2013]. In this work, the authors extend a previous solution for laser-to-camera calibration [Zhang and Pless, 2004] modifying the typical checkerboard calibration pattern and testing different error metrics. Note that the above problem in which a range camera and a 2D laser are calibrated is very similar to the one calibrating a RGB camera and a laser [Zhang and Pless, 2004], where the plane parameters can be easily extracted from the range images [Fernández-Moral *et al.*, 2013b] without the need of any specific calibration pattern. Thus, only a common 3D plane should be sufficient to estimate the extrinsic calibration. As it was demonstrated in [Vasconcelos *et al.*, 2012], three plane-line correspondences are required for that, with the planes having linearly independent normal vectors (they intersect in only one point). Such plane-line correspondences can be obtained through the observation of the same plane from different orientations.

In this section we address the above problem in a probabilistic framework which can be easily generalized to other kind of sensors. For that, the internal calibration of the sensors are assumed to be known. A keypoint of this contribution in reference to previous approaches is the derivation of the approximated maximum likelihood estimation for the calibration, which propagates the uncertainty of the sensor measurements providing the calibration uncertainty itself. This is useful for other problems in the field of mobile robotics like map construction, range-based odometry or simultaneous localization and mapping (SLAM) [Trevor *et al.*, 2012]. Also, the calibration approach presented here is generalized for any geometric configuration of sensors (which may have very divergent fields of view), what is highly interesting for autonomous vehicles.

Contribution

A new method for extrinsic calibration of range cameras and laser rangefinders is presented in this section. This method shares the advantages of the two solutions

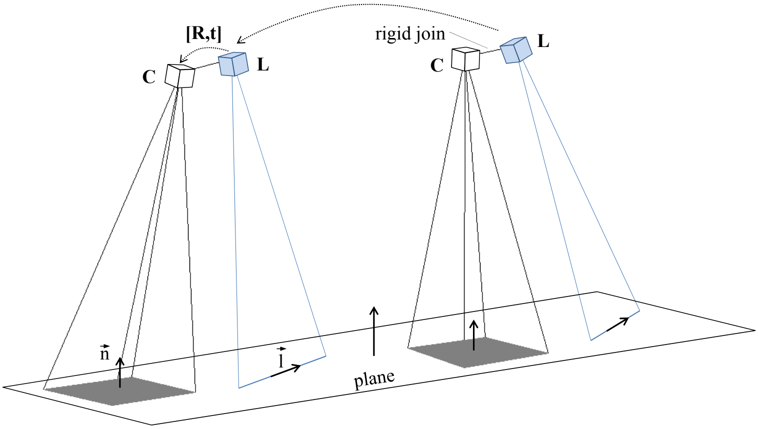


Figure 2.20: A planar surface is observed by two range sensors (a 3D range camera and a 2D laser rangefinder) rigidly joined from different positions. Extrinsic calibration is performed by finding plane-line correspondences from different orientations with respect to the rig's reference system.

presented above, requiring only the existence of an observable plane (e.g. the floor or a wall). Regarding its advantages with respect to previous approaches: it does not require any calibration pattern; it permits to calibrate sensors in arbitrary positions (no need for overlapping fields of view); calibration can be performed rapidly and easily; and finally, the covariance of the resulting calibration is derived according to the sensor model, permitting to propagate the uncertainty of the measurements. The formulation presented in this section can be also applied to other combinations of sensors where 3D planes can be segmented, e.g. RGB and laser. Experimental results are presented in both simulation and real case experiments, showing the advantages of our technique regarding its simplicity with respect to previous solutions.

In the following we give the details of our calibration approach. The formulation of the problem is presented for a pair laser-range camera (section 2.3.3.3). The observability of the problem is studied next. Experimental results are presented in both simulation and real case experiments (section 2.3.3.5). Finally, the conclusions are outlined.

2.3.3.2 Calibration approach

For simplicity, we consider here the problem of calibrating a pair composed of a 3D range camera and a 2D laser rangefinder. The problem of calibrating an arbitrary combination of these sensors can be easily derived nonetheless from the two calibrations formulated above and the new constraints presented in this section. We propose to solve this problem by matching planar and linear features that are seen from different viewpoints. These features are parameterized as described for the solutions presented above to calibrate sets of 2D and 3D range sensors (sections 2.3.1.2 and 2.3.2.2). We make use of large planes in the scene (e.g. the floor or the walls) to establish plane-line correspondences, as shown in figure 2.20. With this strategy we avoid the need of creating an specific calibration pattern for each sensor configuration. Also, neither SLAM nor odometry are needed for calibration [Schneider *et al.*, 2013], avoiding robustness issues and making the procedure much more accessible and easy to use.

Constraint equations

Plane-line correspondences need to be found to constrain the problem of extrinsic calibration. These are the pairs of segmented planes and lines that lie on the same physical plane and are captured in the same time instant with the different sensors. To define the constraints, let's consider a range camera C and a laser rangefinder S which are rigidly jointed, with the laser S relatively located at $[\mathrm{R}|\mathbf{t}] \in \mathbb{SE}(3)$ with respect to C, where the rotation $\mathrm{R} \in \mathbb{SO}(3)$ is represented with a 3×3 matrix and the translation $\mathbf{t} \in \mathbb{R}^3$.

Orientation constraint: A constraint for the relative orientation between the two sensors is inferred from the perpendicularity between the plane's normal vector and a vector lying on the plane, then

$$\mathbf{n} \cdot \mathrm{R}\mathbf{l} = 0 \tag{2.54}$$

being $\mathbf{n}$ and $\mathbf{l}$ the normal vector and the line direction vector observed by the sensors C and S, respectively.

Position constraint: Another constraint involving the relative position between the sensors is stated by comparing the distance from the sensors to the respective plane, which is expressed as

$$d + \mathbf{n} \cdot (\mathrm{R}\mathbf{p} + \mathbf{t}) = 0 \tag{2.55}$$

where d is the observed distance from the plane to the optical center of the depth camera C, and $\mathbf{p}$ is any point on the line segmented by the laser scanner S.

Obtaining plane to line correspondences

Plane-line correspondences need to be found to constrain the problem of extrinsic calibration. These are the pairs of segmented planes and lines that lie on the same physical plane and are captured in the same time instant with the different sensors. Such correspondences are established here by considering all the plane-line combinations of each simultaneous observation as potential matches. Then, after enough good correspondences have been found, outliers can be ruled out robustly using RANSAC [Fischler and Bolles, 1981]. This stochastic procedure calculates the extrinsic calibration (explained in the next section) from a minimum set of 3 plane-line correspondences randomly selected, and searches for the maximum consensus between the whole set of correspondences. This provides good results when most correspondences are correct as it is the case here, since the user drives intentionally the mobile platform or the rig of sensors towards planar surfaces to obtain such correspondences.

This strategy allows us to gather correspondences directly from the streaming data of the sensors (see the video at `http://youtu.be/1VIeP5h_4h4`). The sensors' synchronization effect can be neglected when the motion of the set of sensors is small with respect to the acquisition time. Also, a convergence condition can be set based on the maximum uncertainty (covariance) of the resulting calibration, making the process automatic.

As for the previous calibration problems, the selection of correspondences can be constrained if we know a rough approximation of the sensors relative positions, so that fewer outliers are selected in a first instance. This is a very common situation for most applications, however, it is not applied here for the sake of generality.

2.3.3.3 Problem formulation

From the two constraints presented above, which are modelled as Gaussian distributions, the maximum likelihood estimation (MLE) of the relative pose $[\mathrm{R}|\mathbf{t}]$ is calculated as the maximization of the log-likelihood

$$\underset{[\mathrm{R}|\mathbf{t}]}{\operatorname{argmax}} \left(\ln \prod_{i=1}^{N} p(\mathbf{n}_i, \mathbf{l}_i | \mathrm{R}) \cdot p(\mathbf{n}_i, d_i, \mathbf{c}_i | \mathrm{R}, \mathbf{t}) \right) \tag{2.56}$$

where the likelihood of the observation inferred from the rotation constraint of the i-th plane-line correspondence is approximated as a Gaussian given by

$$p(\mathrm{R}|\mathbf{n}_i, \mathbf{l}_i) = \left(\frac{1}{2\pi(\sigma_i^R)^2} \right)^{1/2} \exp\left(-\frac{(\mathbf{n}_i \cdot \mathrm{R}\mathbf{l}_i)^2}{2(\sigma_i^R)^2} \right) \tag{2.57}$$

being N the number of plane-line correspondences, and σ_i represents the variance of the error function which is derived following appendix C as

$$\frac{1}{(\sigma_i^R)^2} \simeq \frac{1}{\mathbf{n}_i^\top \mathrm{R} \Sigma_{l_i} \mathrm{R}^\top \mathbf{n}_i + \mathbf{l}_i^\top \mathrm{R}^\top \Sigma_{n_i} \mathrm{R} \mathbf{l}_i} \tag{2.58}$$

On the other hand, the likelihood of the observation inferred from the position constraint of the i-th plane-line correspondence is given by

$$p(\mathbf{n}_i, d_i, \mathbf{c}_i | \mathrm{R}, \mathbf{t}) = \left(\frac{1}{2\pi(\sigma_i^t)^2} \right)^{1/2} \exp\left(-\frac{1}{2} \frac{(d_i + \mathbf{n}_i \cdot (\mathrm{R}\mathbf{c}_i + \mathbf{t}))^2}{(\sigma_i^t)^2} \right) \tag{2.59}$$

where σ_i^t is the variance of the error, which after linearisation (see appendix C) is calculated as

$$\frac{1}{(\sigma_i^t)^2} \simeq \frac{1}{\sigma_{d_i}^2} + \frac{1}{\mathbf{c}_i^\top \mathrm{R}^\top \Sigma_{n_i} \mathrm{R} \mathbf{c}_i + \mathbf{n}_i^\top (\mathrm{R} \Sigma_{c_i} \mathrm{R}^\top + \mathbf{c}_i^\top \Sigma_R \mathbf{c}_i) \mathbf{n}_i} \tag{2.60}$$

When the variances are constant with respect to the model parameters, the solution of the MLE in (2.13) coincides with that of the weighted, non-linear least squares problem expressed as

$$\underset{[\mathrm{R}|\mathbf{t}]}{\operatorname{argmin}} \sum_{i=1}^{N} \omega_i^R (\mathbf{n}_i \cdot \mathrm{R}\mathbf{l}_i)^2 + \sum_{i=1}^{N} \omega_i^t (d_i + \mathbf{n}_i \cdot (\mathrm{R}\mathbf{c}_i + \mathbf{t}))^2 \tag{2.61}$$

where the weights ω_i^R and ω_i^t of corresponding constraints of the i-th plane-line correspondence are given by

$$\omega_i^R = \frac{1}{(\sigma_i^R)^2} \; , \; \; \omega_i^t = \frac{1}{(\sigma_i^t)^2} \tag{2.62}$$

This problem is reformulated using Lie algebra (see appendix B) to represent the poses with a minimal parametrization on a manifold. For that, the rotations are represented as the composition of a guessed rotation and a rotation increment represented with the exponential map ($e^{\mu_j}\mathrm{R}_j$), with the rotation increment $e^{\mu_j} \in \mathbb{SO}(3)$. The translations are also represented as the sum of a guessed translation plus an increment ($\mathbf{t}_j + \Delta\mathbf{t}_j$), both in $\mathbb{R}^3$. The resulting non-linear least squares problem is solved iteratively using Levenberg-Marquardt as in section 2.3.1.3 (eq. 2.20), where the Hessian H (a 6-dimensional symmetric matrix) and the Gradient g (a 6-dimensional column vector) of the cost function are calculated from

$$H = \sum_{i=1}^{N} J_i^\top \omega_i J_i \; , \; \; g = \sum_{i=1}^{N} J_i^\top \omega_i r_i \tag{2.63}$$

where N is the number of constraints of this optimization, being r_i the residual and J_i the Jacobian for each constraint. The Jacobian J_i corresponding to the constraint from eq. 2.54 and its residual are calculated as

$$J_i = [(\mathbf{n}_i \times (\mathrm{R}_j \mathbf{l}_i))^\top, \, 0, \, 0, \, 0] \; , \; \; \mathbf{r}_i = \mathbf{n}_i \cdot R_j \mathbf{l}_i \tag{2.64}$$

The Jacobian and the residual corresponding to the constraint from eq. 2.55 are given by

$$J_i = [(\mathbf{n}_i \times (\mathrm{R}_j \mathbf{c}_i))^\top, \, n_i^\top] \; , \; \; \mathbf{r}_i = d_i + \mathbf{n}_i \cdot \mathrm{R}\mathbf{c}_i \tag{2.65}$$

This procedure requires an initial guess for the relative pose which may be provided according to the sensor rig design. The covariance of the resulting calibration corresponds to the Hessian of the above problem [Fernández-Madrigal and Claraco, 2013].

2.3.3.4 Observability

As it was described in the previous section, each plane-line correspondence imposes two new constraints between the pair of sensors: one for the relative rotation and one for the relative translation. Thus, we need at least 3 measurements from linearly independent observations (i.e the normal vectors of the observed planes are linearly independent) to compute the relative position between a range camera and a laser rangefinder. Following the procedores of the previous sections, this is demonstrated from the analysis of the Fisher Information Matrix (FIM) of the estimation problem above, which coincides with the Hessian matrix of eq. 2.63 (see the appendix D). Then, when the FIM is singular (its rank is not full), the information provided is not sufficient and the problem cannot be solved.

By analysing the translation block FIM_t of the Fisher Information Matrix

$$FIM_t = \sum_{i=1}^{N} \mathbf{n}_i \omega_i \mathbf{n}_i^\top = XWX^\top \tag{2.66}$$

where n_i is the normal vector of the plane i as seen from the range camera, $X = [\mathbf{n}_1|\mathbf{n}_2|\dots|\mathbf{n}_N]$ a $3\times N$ matrix, and W is a $N\times N$ diagonal matrix with the weights ω_i in its i-th element. The $rank(FIM) = 6$ only if FIM_t has full rank ($rank(FIM_t) = 3$), what is the case when there are 3 observed normal vectors that are linearly independent

$$rank(FIM_t) = rank(X) = 3 \tag{2.67}$$

The same theory applies for the rotation block FIM_R, which depends on both types of constraints. Considering only the orientation constraints

$$FIM_R = \sum_{i=1}^{N} (\mathbf{n}_i \times (\mathrm{R}\mathbf{l}_i))\omega_i(\mathbf{n}_i \times (\mathrm{R}\mathbf{l}_i))^\top = YWY^\top \tag{2.68}$$

where $Y = [\mathbf{n}_1 \times (\mathrm{R}\mathbf{l}_1)|\mathbf{n}_2 \times (\mathrm{R}\mathbf{l}_2)|\dots|\mathbf{n}_N \times (\mathrm{R}\mathbf{l}_N)]$ is a $3\times N$ matrix. Assuming that the plane-line correspondences are correct and that the initialization for the rotation is in a close neighbourhood of the solution, then $rank(FIM_R) = 3$ always when the condition of eq. 2.67 is met. Therefore, the problem is observable if and only if 3 planes with different orientations are observed.

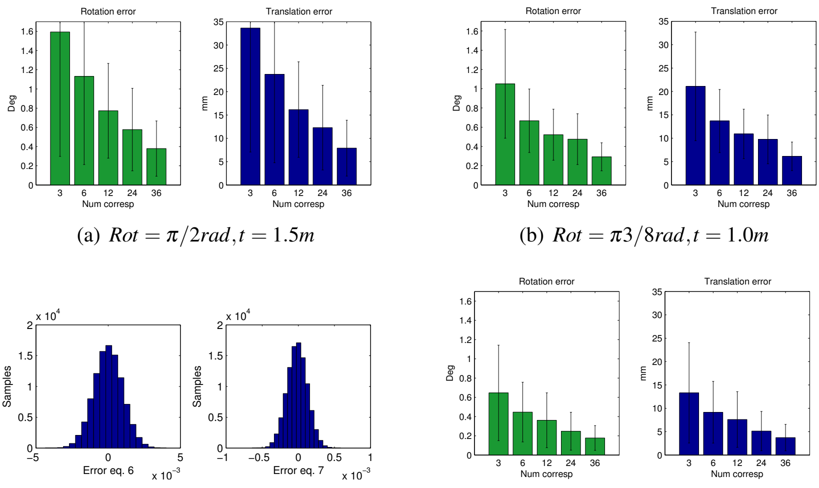
From this result, we see that the pair of sensors can be calibrated from a single observation when there are 3 perpendicular planes that are visible by both sensors. This can be achieved by observing the corner of a room or a building.

2.3.3.5 Experiments

A number of experiments has been carried out to validate the present approach. Both simulation and real cases experiments are presented in this section.

Simulation

The accuracy of our method is evaluated with some simulation experiments since a groundtruth for the relative poses of the range sensors of our mobile platforms is not available. For that, a large plane is observed from different sensor positions and orientations of a rig containing a laser and a 3D range camera. The rig is positioned



(a) $Rot = \pi/2rad, t = 1.5m$

(b) $Rot = \pi 3/8rad, t = 1.0m$

(c) $Rot = \pi/4rad, t = 0.5m$

(d) $Rot = \pi/8rad, t = 0.2m$

Figure 2.21: Average errors in rotation (measured as the module of the difference of the rotation vectors) and translation (module of their Euclidean distance) for different numbers of plane-line correspondences and different relative poses between the sensors. 200 trials are performed for each set of parameters.

at different distances from the plane in a range between 1 m and 3 m, and it is rotated around some random axis to gather measurements in different directions, as shown in figure 2.20. The sensors are modelled according to the parameters of the depth camera of a Kinect and a Hokuyo UTM-30LX rangefinder. The observations are modelled following these sensor specifications, where the error in the measurements is modelled as unbiased Gaussian noise, for the Kinect, the σ_K depends on the true depth and the incidence angle of the ray of observation with the 3D plane, and the σ_L of the laser is set constant. Plane and line features together with their covariances, are extracted from such observations as explained in section 2.3.3.2. Data association is not tackled in this simulation as all the measurements come from the same plane.

The rotation and translation are then estimated from the simulated observations for different conditions: varying number of correspondences, different incidence angle of the sensors optical axis and the plane, and different relative transformations between the sensors. The initial rotation is set to the a random pose at an angle of $[0, \pi/4]$ radians and a distance of $[0, 1]$ m of the true relative pose. The average errors with respect to the true pose for the rotation and the translation are obtained for a total of 200 trials for every set of conditions, see figure 2.21. From this experiment, we observe the trend that the average errors in rotation and translation decrease as the number of correspondences grows, as more measurements help to reduce the covariance. Also, we see that the further apart the relative poses between the sensor are,

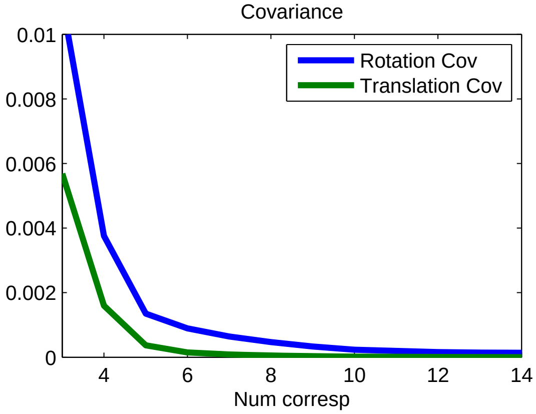


Figure 2.22: Maximum eigenvalues of rotation and translation covariance with respect to the number of plane-line correspondences.

the bigger the error. This makes sense, since sensors far away will have few common observations.

Another interesting point is to evaluate the convergence of the algorithm, so that plane-line correspondences are gathered until a threshold for the calibration's uncertainty is reached. For that we draw the maximum eigenvalue of the calibration's covariance as the number of correspondences grows for both rotation and translation, see figure 2.22. We see how the covariance decreases asymptotically to zero as more correspondences are gathered. The threshold to stop calibration can be set easily by the user according to his needs.

Real data

We present here the results of a real case experiment to illustrate the precision of the calibration method. We have calibrated a Hokuyo UTM-30LX laser rangefinder with respect to a Kinect RGB-D camera. Both sensors are mounted on a Pioneer PatrolBot as is shown in figure 2.23. The sensors' synchronization effect is neglected in this test since the laser scans and range images are captured at a minimum frame rate of 30 Hz, and the robot is moved at low speed.

The accuracy of the resulting calibration from our method cannot be evaluated directly since we do not have a groundtruth for the sensors relative positions. Thus, following the works in the literature, we provide some qualitative results based on the residual errors. In table 2.7 we show an example of how the average residual error is reduced when raising the number of plane-line correspondences. The residual error in rotation and translation are measured in a dataset containing 1000 correct correspondences for several calibrations with varying number of correspondences. In this experiment we placed the robot near a wall, so that correspondences could be ac-

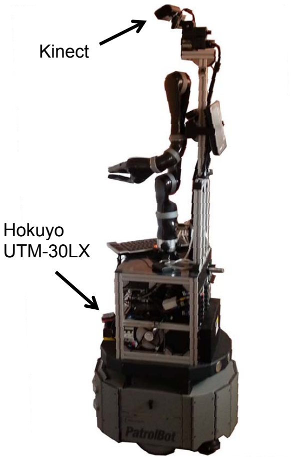


Figure 2.23: Robot which has a 3D range camera and a 2D laser rangefinder.

quired quickly in different directions (from the wall and the floor) as it is shown in the video `http://youtu.be/1VIeP5h_4h4`. In this situation, calibration is performed in a few seconds with around 12-100 plane-line correspondences.

Table 2.7: Residual errors for different calibrations using a different number correspondences.

Correspondences	*Av rot error (deg)*	*Av trans error (cm)*
3	1.31	3.17
10	0.71	2.01
20	0.65	1.52
40	0.60	1.34
80	0.55	1.33

2.3.3.6 Discussion

A new methodology for calibrating the extrinsic parameters of a rig of range sensors has been presented in this section, namely a 3D range camera and a 2D laser rangefinder. The method relies on the matching of plane and line observations from the different sensors. The extrinsic calibration problem is solved in a probabilistic framework that takes into account the uncertainty in the measurements of the sensors, and provides also the uncertainty of the resulting calibration. No constraints are put on the position of the cameras, where the only requirement for the system is that there is a planar surface that can be observed simultaneously by all the sensors. This approach can be easily extended to other types of sensors always when a plane can segmented from the scene. The observability conditions and some insight in the convergence of the problem is presented. With our method, performing calibration becomes very fast and easy for the user, avoiding problems of previous solutions which rely either on calibration patterns or trajectory estimation methods. The method has been tested in simulation and real case experiments validating the claimed features of our proposal.

2.4 Conclusions

This chapter reviews some relevant methods for calibrating different combinations of sensors which are usually employed in mobile robotics. A new methodology for calibrating combinations of range sensors has been presented. Three different problems are tackled here to calibrate: a) a set of 2D range scanners, b) a set of 3D range cameras, and c) a 2D and a 3D range sensors. Different methods are presented for each problem which can be easily combined to calibrate any combination between them. All these methods are based on the observation of planar surfaces from structured environments. The proposed methods have a number of advantages with respect to previous approaches, namely: they can be applied to any geometric configuration of the rig of sensors; they do not need external information (calibration patterns, special landmarks, etc.); they are easy to apply, performing calibration in a few seconds; and they provide the uncertainty of the resulting calibration.

The method to calibrate combinations of 2D range scanners are mainly interesting in the context of autonomous cars, where most prototypes in the literature make use of several 2D LRFs. On the other hand, the calibration of 3D range cameras looks more interesting for indoor robotic applications, providing large field of view at low cost of the sensor system. The observability of each one of the proposed methods is analysed, showing that a single observation of the rig may be enough to calibrate the sensors.

Regarding the experimental evaluation carried out here, the presented techniques are not compared to previous approaches in the literature since they cannot be compared in the same conditions. Still, an interesting aspect would be to measure the accuracy of the techniques presented in the real experiments to set some bounds on their applicability and to provide some quantitative information for future comparisons. Obtaining such results would imply a complex and expensive procedure to obtain some kind of groundtruth that was not available during the research of this thesis. Therefore, this aspect stays as a possible line of future research.

Chapter 3
Plane-based maps for fast localisation and place recognition

Abstract

Different kinds of maps have been used in mobile robotics for self-localization, navigation or scene reasoning. A new type of map is proposed in this chapter which is based on the registration of planar surfaces, and which is extremely compact in comparison with previous approaches. This world representation is organized in a graph where the nodes represent the planar patches and the edges connect neighbouring planes. This map structure allows to work efficiently with local regions by selecting subgraphs of neighbour planes that can be quickly compared for real-time place recognition and scene registration. For that, an interpretation tree is employed to find a candidate match between two subgraphs by searching the best combination of planar patches that fulfils a series of geometric and radiometric constraints. Such a strategy permits working with partially observed and missing planes, offering invariance to viewpoint and robustness against changes in the scene. The proposed approach constitutes an efficient way to solve loop closure detection and scene registration, working satisfactorily even when there are substantial changes in the scene (lifelong maps).

3.1 Introduction

As discussed in the first chapter of this thesis, compact scene representations are highly interesting for efficient SLAM operation, among other utilities. Take into account that a mobile robot can gather information from the environment continuously like we humans do. Besides, lightweight mapping solutions are desired when the computation burden is limited, like in wearable applications or in robotic solutions which integrate other demanding functionalities (e.g. task planning and execution, or events anticipation). In such a context, efficient memory and processing mapping strategies are highly advantageous.

Different mapping strategies have been proposed in mobile robotics depending on the purpose of the robot and the available sensors. A popular alternative to represent the world when using RGB-D sensors (like Microsoft Kinect) is through point clouds, where coloured points are rendered to the map according to the sensor's pose [Kerl *et al.*, 2013a]. Such a representation offers a high degree of detail, achieving nice visualizations of the scene that can be employed in several fields apart from mobile robotics, like scene modelling, augmented reality or video games. However, the compactness of this representation is not suitable for applications with more limited memory and processing resources.

The problem of re-localization is a good example where compact maps are highly interesting. The ability to quickly recognize a previously visited place is a major problem in mobile robotics since, among other things, it allows to accomplish topological localization and loop closure detection in SLAM. In contrast to the typical localization problem in SLAM where the robot tracks a local map around its last position, a re-localization algorithm will check generally a much larger part of the map, requiring more computation. The map being checked for that will depend on the uncertainty of the robot trajectory. For instance, in the case of robot awakening problem, where information about the current location is not available, the whole map will need to be checked. The main issue here is how to describe the generally big amount of information present in the scene in order to recognize a place in a robust and affordable way when it is visited again.

This chapter proposes a compact plane-based representation of the scene from RGB-D data that we name PbMap (Plane-based Map) [Fernández-Moral *et al.*, 2013b], which is specially useful for place recognition and re-localization. The PbMap stores only the planar skeleton of point clouds, and thus, it avoids the redundancy of information in them, where coplanar points are represented compactly by a convex hull. This representation was partly inspired by the CAD models, that encode metric information in a compact fashion and still, they can be easily interpreted by humans despite the lack of information about scale, texture, etc. (see figure 3.1.a). Also, a plane constitutes a higher level feature of semantic information with respect to 3D points, as planes normally correspond to meaningful objects (e.g. a wall, or a door) and a few planes can compose a semantically meaningful object (e.g. a table, a desktop, etc.), that can be exploited for semantic mapping [Ruiz-Sarmiento *et al.*, 2014]. On the other hand, this model loses descriptiveness with respect to point clouds since

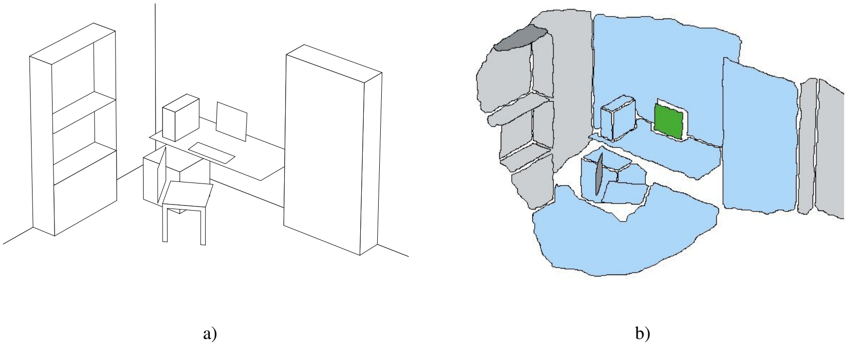


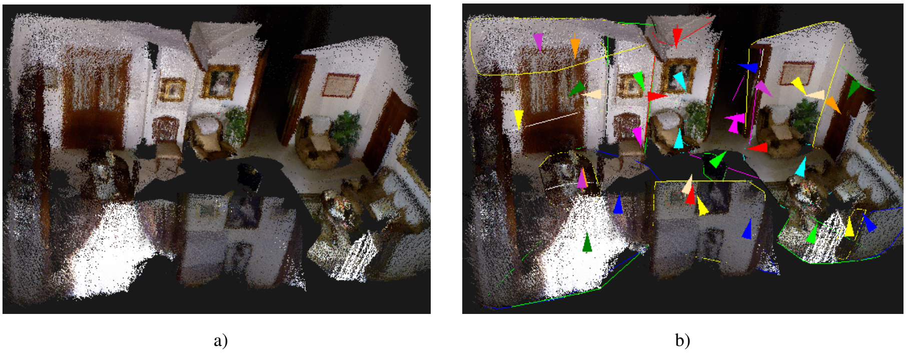
Figure 3.1: Example of a typical scene that can be represented with planar patches. a) CAD model b) PbMap of the scene where a local neighbourhood of planes is represented, which is defined by a reference plane (green), and includes its closest planes up to a distance threshold of 1 m (blue).

the information from non planar areas is discarded. Thus, our approach assumes that there are enough planar patches, and then structured indoor scenes are more adequate for this representation.

A PbMap is organized as an annotated graph where each node corresponds to a planar patch (described by simple features: normal vector, centroid, area, colour, etc.) and the edges connect neighbour patches according to their proximity and co-visibility. These planar patches (or planes, for short) can be extracted in real-time from the range video streaming provided by a hand-held range camera or a RGB-D sensor. Such planes are integrated into the map in their respective poses according to the sensor location. The sensor pose can be obtained in different ways, in this chapter it is estimated from the sensor observations using an odometry algorithm [Steinbrücker *et al.*, 2011]. The use of odometry only for constructing our maps implies that our representation is topological in nature, but note that re-localization and place recognition do not require fully consistent maps to work.

Place recognition in PbMaps is addressed here as a problem of matching subgraphs, which represent the so-called “contexts of planes”. For loop closure detection, the subgraphs representing the current observed planes are compared with other ones from the PbMap. Such subgraphs can be defined by one reference plane together with their closest neighbours, up to a distance threshold (see figure 3.10). For solving the graph matching problem we rely on an interpretation tree [Grimson, 1990] that exploits the geometric and radiometric characteristics of the planes and their relative positions to generate a set of constraints that guide efficiently the search.

A registration method for aligning two matched places is also presented here. This registration is applied after graph matching to check the consistency of the matched places. Thus, it improves the robustness of the recognition since a good registration is required to accept the validity of the matched place. This consistency test computes



a)

b)

Figure 3.2: Plane-based representation. a) Point cloud representation of a living room. b) Point cloud representation with the segmented planar patches superimposed.

the adjustment error together with the relative pose of the sensor with respect to the place recognized, thus providing an estimate for the localization.

Experimental results are provided demonstrating the effectiveness of our method for recognizing and localizing places in a dataset composed of several home and work-place scenes (e.g. offices, living rooms, kitchens, bedrooms, corridors, etc.). The proposed registration has also been tested for omnidirectional RGB-D images, showing a performance (accuracy *vs.* computation) suitable for odometry and SLAM (this is addressed in chapter 5). Also, in order to test the concept of "lifelong map" [Konolige and Bowman, 2009] we also show how the recognition is affected by the fact that the scene suffers some changes.

3.1.1 Related works

Mapping

Different mapping solutions have been presented depending on the characteristics of the problem. One of the first ones being employed were the 2D occupancy grid maps, in which the space is organized in cells which keep the probability to be occupied [Moravec and Elfes, 1985]. This representation has been very popular for navigation of wheel robots equipped with 2D laser rangefinders moving on a plane [Dellaert *et al.*, 1999]. A 3D version of this map makes use of cubic cells (or voxels) [Lozano Albalate *et al.*, 2002]. The main limitation of this kind of maps comes from its storage requirements which becomes a problem in large scenarios. *Octomaps* are a way of organizing such information where each voxel is subdivided in 8 sub-voxels depending on the variability of neighbouring cells and the precision required [Wurm *et al.*, 2010].

An alternative to the grid representations are the maps based on distinctive features or landmarks that can be extracted from a textured scene with one or more cameras [Se *et al.*, 2002]. In this line, most works in the literature have focused on point-based maps, where different invariant point descriptors have been used, from the popular SIFT [Lowe, 2004], or SURF [Bay *et al.*, 2008], to patch descriptors like in [Davison and Murray, 2002]. Feature maps from range data have also been proposed more recently with the spread of 3D range sensors [Rusu *et al.*, 2008].

Dense point maps have also become very popular in the last years with the arrival of low cost RGB-D sensors like Microsoft Kinect. These representations offer a fair degree of detail that permits creating point cloud models with nice visualizations which have many applications beyond robotics. However, the scalability of such maps becomes an issue for many mobile applications, especially for large-scale SLAM operation. A common approach to reduce data redundancy is by using *keyframes*, in a similar fashion to previous monocular SLAM approaches [Klein and Murray, 2007]. In this line, recent combinations of the state-of-the-art techniques exploiting the power of modern computers have resulted in impressive 3D maps from hand-held RGB-D sensors [Kerl *et al.*, 2013b]. Continuous surface representations have also been proposed based on polygonal representations [Lafarge and Alliez, 2013], and piecewise continuous radial basis functions (RBFs) [Carr *et al.*, 2001]. Such models are useful for scene reconstruction and augmented reality, but are not suitable for fast localization in comparison to other approaches.

Besides metric mapping strategies, pure topological representations have been employed when accurate metric localization is not required [Ulrich and Nourbakhsh, 2000]. Also, semantic information maps have been presented as a way to integrate useful knowledge about the objects and the environment [Galindo *et al.*, 2005]. Hybrid maps integrating metric and topological information have also been proposed to deal with large and complex scenarios [Blanco *et al.*, 2009a]. These kind of maps are discussed in the following chapter of this thesis.

Compact scene representations are interesting for efficiency and scalability. In general, the more compact the model is, the less information and weaker description it offers. The balance between compactness, and accuracy (or descriptiveness) of the model must be taken into account according to the application. For example, this balance is adjusted with the size of the cell in a gridmap [Elfes, 1989], with the number of features in keypoint maps [Dissanayake *et al.*, 2001] and point clouds [Rusu *et al.*, 2008], or with the size of discretization for piecewise continuous RBFs representations [Carr *et al.*, 2001].

In this thesis we address the creation of a compact representation which only contains planar patches. This is the first work, to the best of our knowledge, in which planes are integrated to a map at a high frame rate with unconstrained camera movement. This plane-based representation constitutes a useful framework for object detection and grasping [Klank *et al.*, 2009], visual servoing [Cowan and Koditschek, 1999] and Manhattan-like modelling [Furukawa *et al.*, 2009]. Such planar patches can be efficiently extracted from depth images [Poppinga *et al.*, 2008], [Holz and Behnke, 2013]. In the context of SLAM, some approaches have already used planar surfaces as the only map features [Weingarten and Siegwart, 2006; Trevor *et al.*, 2012], or along with other features (like point features) [Chekhlov *et al.*, 2007; Gee *et al.*, 2008; Martinez-Carranza and Calway, 2012]. Our approach differs from the above in the graph-based representation that we employ to characterize the scene, which permits to take into account the relations between neighbouring planes to perform fast and robust place recognition.

Place recognition

The problem of place recognition has been addressed previously from different perspectives using different sensors, from laser-range finders to different kind of cameras (e.g. consumer cameras, stereo vision, omnidirectional imaging and range cameras). Most solutions for this problem employ intensity images as input data. Appearance based methods, applied previously for object recognition [Murase and Nayar, 1995], have been largely studied in this sense. We can distinguish two kind of approaches here depending on whether the scene is described with local descriptors (local appearance) or using a global descriptor (global appearance). Local appearance methods, like the popular bag-of-words (BoW), represent the images as an unordered set of visual features (words), that are generally collected in a dictionary in a previous stage. Such a dictionary is built by clustering similar descriptors to create visual words that are repeatable. Then, different places can be recognized by classifying the images according to the frequency of their words [Sivic and Zisserman, 2003; Csurka *et al.*, 2004]. A relevant example which makes use of bag-of-words is the work of [Cummins and Newman, 2008], which recognizes places quickly by capturing the fact that certain combinations of appearance words tend to co-occur. Also, [Angeli *et al.*, 2008] presented an incremental, real-time system to detect loop-closures within a Bayesian filtering framework. The orderless bag-of-words technique is extended to take into account geometric correspondence in [Lazebnik *et al.*, 2006], improving the recognition performance.

In contrast to the solutions above, global appearance methods describe the scene as a whole. In this line, the method presented in [Ulrich and Nourbakhsh, 2000] makes use of an omnidirectional camera to find the location in a topological map employing maximum likelihood estimation to match the current image with a database of images acquired beforehand. Contemporary with this, the work of [Kröse *et al.*, 2001] presented a probabilistic localization method that employs linear image features extracted using Principal Component Analysis. A context-based vision system for place and object recognition was presented in [Torralba *et al.*, 2003] which identifies familiar locations employing a low-dimensional global image representation. In [Oliva and Torralba, 2001], a holistic representation of the scene's spatial envelope is proposed. This work is extended in [Oliva and Torralba, 2006] introducing a scene centred image global descriptor to find places based on configuration of spatial scales. This method has common aspects with ours since it relies on the global scene layout and it can be combined with local image analysis (like BoW) to constrain the search space and to improve performance. An important difference however, is that our technique is not restricted to work with individual images, and thus, it can integrate several sensor observations in the same scene description, providing inherently a high invariance to viewpoint.

There exist other methods in the literature which also address the place recognition problem from range data: the work of [Bosse and Zlot, 2008] presents a solution for place recognition which employs distinctive keypoints from 2D lidar observations. This approach is extended to 3D laser point clouds in [Bosse and Zlot, 2010]. The work in [Granström *et al.*, 2011] also employs range data to extract features that capture important geometric and statistical properties to detect loop closures. Our approach differs from the ones above in different aspects: a) our method exploits contextual information of nearby planes, b) it does not require a training step, and c) it describes the scene in a more continuous way with a plane-based representation which is useful beyond place recognition (e.g. scene modelling).

The recent availability of low cost RGB-D sensors has given rise to new approaches for the problem. In [Biswas and Veloso, 2012] the depth image from a Kinect sensor is used for localization and navigation. This approach extracts planar regions to reduce the computation load of using dense point clouds, and projects the points and planes in a 2D vector map to localize the robot and to avoid obstacles in previous 2D-range maps. However, this approach does not exploit the implicit description of planar regions and neglects important 3D information in the scene description. The method proposed in [Koppula *et al.*, 2011] segments the scene and automatically labels these segments using a machine learning approach that takes into account local visual appearance and geometric cues, together with contextual information. Our method resembles the one above in the use of geometric information and proximity to establish the context regions, however, this method is focused on scene understanding, and therefore, requires a training stage, while ours aims to recognize previous places and does not need any off-line preparation.

3.1.2 Contribution

We propose a highly compact map representation of the scene based on planar patches that can be built online from the streaming data of a RGB-D sensor. The novelty of our representation is that such planes are described with a compact geometric and radiometric descriptor, and that such planes are integrated in a graph representation which stores the “topological” relations between planes, which permits quick checking for similar place descriptions. This new representation is highly efficient for recognizing and registering places, having the following advantages:

1. the description of the scene through a PbMap is very compact, requiring little memory and reducing the computational cost of search operations;

2. it is robust to changes of viewpoint since the scene planes can be detected from very different poses, and the context of planes to match can be chosen with flexibility (it is not restricted to single-image discretization);

3. it tolerates reasonably well changes in the scene, and therefore is adequate for the so called “lifelong maps”, i.e. maps that are still valid after the scene changes. This characteristic particularly holds for indoor scenarios, where the

most visible and larger planes (i.e. walls, floor, ceiling, bigger furniture, etc.) are normally persistent over time, while other smaller objects (e.g. chairs, a laptop, a backpack) are more likely to be moved or even disappear.

An implementation of PbMaps and the registration technique which is used for localization and place recognition is made available at `http://www.mrpt.org/pbmap`.

The rest of this chapter is structured as follows: first, we address the plane segmentation problem, which needs to be solved prior to the map construction. Next, a PbMap is described as a set of geometric and radiometric characteristics which are encoded in a graph representation (section 3.3). A compact colour descriptor for planar patches is presented next (section 3.4). The construction of a PbMap and its update from new observations of a range or RGB-D sensor is described in section 3.5. Then we show how the PbMap is used to search for similar scenes by matching subgraphs (section 3.6). The experiments and their results are presented next (section 3.7). Finally, we expose the conclusions of our work.

3.2 Plane segmentation

The problem of plane segmentation has been long studied in computer vision. Different techniques are applied for range and visual data respectively. For the case of visual data, planes can be inferred from vanishing points and lines [Hartley and Zisserman, 2003; Košecká and Zhang, 2005; Micusik *et al.*, 2008], from local and global features learning [Hoiem *et al.*, 2007; Saxena *et al.*, 2009; Haines *et al.*, 2013], or from 3D point features extracted with a moving camera [Gee *et al.*, 2008]. The stochastic technique of Random Sampling and Consensus (RANSAC) [Fischler and Bolles, 1981] has become a standard tool for robust generation of plane hypothesis [Zuliani *et al.*, 2005; Martinez-Carranza and Calway, 2012], which can be further checked with homography restrictions [Argiles *et al.*, 2011]. Apart from RANSAC, other approaches like those based on seed initialization and votation have demonstrated good performance to obtain planar patches from a sparse point-based model [Martínez-Carranza and Calway, 2010].

Methods based on the Hough transform are well known for segmenting lines and circles in images. This kind of technique can be also applied to segment specific patterns from point clouds, like planes [Vosselman *et al.*, 2004; Borrmann *et al.*, 2011]. Hough transform methods, like RANSAC, have the disadvantage to cluster together unconnected regions, since the only restriction imposed is the plane equation. This makes difficult to distinguish patches from different objects when they lie nearly in the same infinite plane. Also, both techniques do not take advantage of the spatial organization of the data when it is available.

Unlike regular intensity images, range images directly provide geometric information that can be exploited to segment planar patches. Region growing techniques can be efficiently applied in range images since the data is already organized (i.e. the neighbour points of a given pixel can be directly accessed with the pixel index:

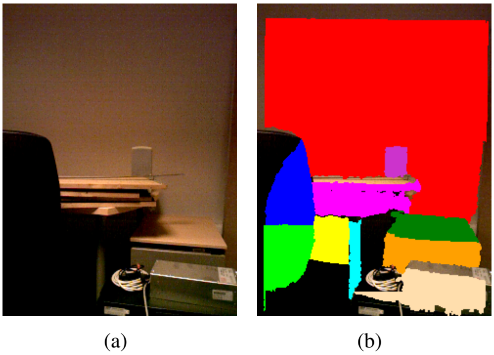

(a) (b)

Figure 3.3: Segmentation of planar surfaces from a 160x120 depth image. a) RGB image of the corresponding view, where the segmented are represented with different colours in b).

row and column) [Hoover *et al.*, 1996; Poppinga *et al.*, 2008]. Merging strategies are often applied after a first segmentation stage to improve the results, so that contiguous patches with a similar plane equation are integrated together [Holz and Behnke, 2013]. In this thesis, we have followed the work of Holz and Behnke [Holz and Behnke, 2012] to segment planar patches from range images. The segmentation is carried out with region growing by limiting the maximum curvature and depth difference between neighbouring pixels. Thus, the first step is to calculate the surface normals and curvature estimates of the image pixels [Holzer *et al.*, 2012]. Next, a bilateral filter is applied to smooth both depth measurements and normal estimates [Weiss, 2006]. These methods are implemented in the Point Cloud Library (PCL)[1] [Rusu and Cousins, 2011]. In order to improve the segmentation in a longer range of distances, a varying threshold for the maximum change of curvature has been introduced to compensate for the measurements noise, which increases quadratically with depth.

The planar patches are segmented from 160×120 range images in real time, offering a rate above 30 Hz. The results obtained with higher resolution images in terms of segmented planes are almost identical, while the computation time scales linearly with the image size. At lower than 160×120 resolutions, the number of segmented planes starts to drop. The segmentation stage is the main computation load for the PbMap construction, taking 13 ms in average per frame.

3.3 Formal definition of a PbMap

A plane-based map (PbMap) is a representation of the scene as a set of 3D planar patches. It is organized as an annotated, undirected graph G, where each node represents a planar patch and the edges connect co-visible neighbour planes, that is, an

[1] The Point Cloud Library is publicly available at `http://www.pointclouds.org/`

edge connects two patches when they have been observed in the same image and the distance between their closest points is under a threshold (see figure 3.1.b).

Each plane $P_i \in G$ is described by a set of geometric features:

- $\mathbf{c}_i$ the centroid,
- $\mathbf{n}_i$ the normal vector,
- $\mathbf{d}_i$ the principal vector (i.e. dominant direction of the plane)
- e_i the elongation,
- a_i the area,
- $\mathbf{H}_i$ a set of points defining the patch's convex hull.

The plane centroid $\mathbf{c}_i$ is calculated as the average of the plane's 3D points $\mathbf{p}^i_j$

$$\mathbf{c}_i = \frac{1}{m} \sum_{j=1}^{m} \mathbf{p}^i_j \tag{3.1}$$

The normal vector $\mathbf{n}_i$ and the principal vector $\mathbf{d}_i$ are the eigenvectors corresponding to the smallest and the largest eigenvalues of the covariance matrix M, respectively.

$$\mathrm{M} = \sum_{j=1}^{m} (\mathbf{p}^i_j - \mathbf{c}_i)(\mathbf{p}^i_j - \mathbf{c}_i)^\top \tag{3.2}$$

The elongation e_i is computed as the ratio between the two largest eigenvalues of M, and the area a_i is computed from the convex hull $\mathbf{H}_i$. All these features are obtained from the plane segmentation and map construction stages, that is, they are set when a plane is initialized and are updated when such plane is re-observed. The patch's convex hull serves besides to calculate the minimum distance between two patches. Besides this geometric description, colour information may be added to the geometric descriptor if it is available. This is comprehensively addressed in the next section.

3.4 Compact colour descriptor

In this section we investigate how to incorporate colour information to a PbMap to improve its descriptiveness while maintaining the model compactness, which is essential in real-time applications [Fernández-Moral *et al.*, 2014a]. The context of this research is that of matching planar patches for real-time tasks like scene registration or place recognition, which involves extensive search for patch correspondences. Thus, selecting a colour descriptor involves the non-trivial aspect of maintaining a trade-off between distinctiveness, compactness, and computational cost.

This problem of finding a colour descriptor for planar patches was posed by [Pathak *et al.*, 2012] in the context of registering 3D range scans, where the authors adopted a hue based histogram to increase the efficiency of registration. In this section we explore the idea of finding a descriptor based on the dominant colour for several reasons: first, most planes present in indoor environments have a dominant colour; second, the dominant colour is more robust to the partial observation of planes; and finally, the efficiency of on-line back-end processes for loop closure or place recognition will benefit from a more compact, fast to compare descriptor.

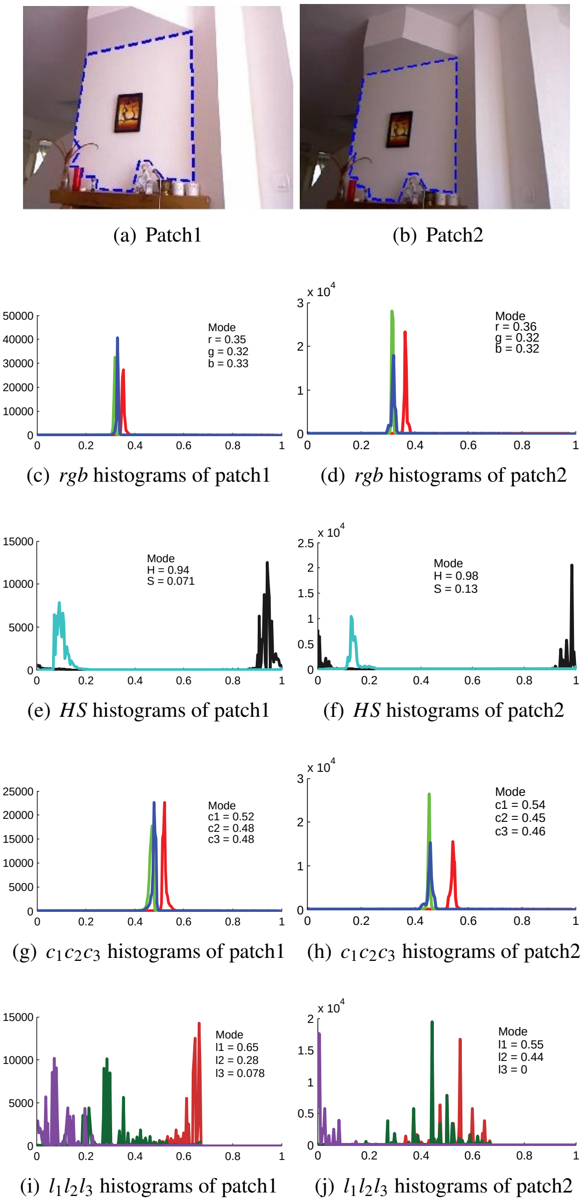
In order to obtain such a descriptor, we study different colour representations and radiometric features, looking for invariance to illumination, point of view and partial occlusion. We propose a colour descriptor based on the patch dominant colour in normalized *RGB* space, which provides the best balance between distinctiveness and compactness. This descriptor is compared with the hue histogram descriptor [Pathak *et al.*, 2012], which was previously proposed for a similar problem.

3.4.1 Colour information for patch matching

We address the problem of finding the simplest colour descriptor for a planar patch focused on the problem of patch matching. This descriptor must be highly invariant to viewpoint, lighting conditions and partial occlusion, and also, it must be efficiently calculated. Note that the utility of this descriptor is not to identify unequivocally planar patches, but to prune the search space of possible matches by adding radiometric information to the geometric attributes of a planar model.

A common solution for matching planar patches is that of maximizing the photoconsistency between them [Argiles *et al.*, 2011]. The main limitation of this strategy is that maximizing the photoconsistency is prohibitively expensive for many applications, especially when there is not a good initial estimation of the registration (e.g. loop closure detection). Closer to our work are those that describe the patch radiometric information through its histogram [Hafner *et al.*, 1995], [Swain and Ballard, 1991]. In this line, [Pathak *et al.*, 2012] posed recently the problem that we consider in this section, showing that colour information can be exploited to increase the efficiency of 3D scan registration. A well illuminated scene is assumed in that work, where the authors adopt a hue based histogram with 2 extra bins to keep saturated values (black and white). Here, we also take into account the fact that many planar patches have a single colour, so that the histogram contains redundant information.

In contrast to the works above, we propose to describe the patch with its dominant colour. A similar strategy is used in video compression [Manjunath *et al.*, 2001] to define blobs having the same colour. In this way the descriptor storage and the computation of distances are reduced to a minimum. This is important in a number of problems where many match combinations have to be checked in real-time. In order to select such a descriptor we need to address some issues: first, the selection of the colour space which offers the best suitability to obtain an invariant and distinctive dominant colour (subsection 3.4.2); second, to define the way this dominant colour is extracted (subsection 3.4.3); and third, to adapt the descriptor for cases where the dominant colour is not reliable (subsection 3.4.4).



(a) Patch1 (b) Patch2

(c) *rgb* histograms of patch1 (d) *rgb* histograms of patch2

(e) *HS* histograms of patch1 (f) *HS* histograms of patch2

(g) $c_1c_2c_3$ histograms of patch1 (h) $c_1c_2c_3$ histograms of patch2

(i) $l_1l_2l_3$ histograms of patch1 (j) $l_1l_2l_3$ histograms of patch2

Figure 3.4: Two patches of the same plane extracted from different views with their histograms in the colour spaces *rgb*, *HS*, $c_1c_2c_3$ and $l_1l_2l_3$ below. We can appreciate that the histograms in *rgb* and $c_1c_2c_3$ exhibit more clearly unimodal, and more stable distributions.

Table 3.1: Formulation of several colour spaces from the RGB data.

Colour space	*Formulation*
rgb	$r(R,G,B) = \frac{R}{R+G+B}$ $g(R,G,B) = \frac{G}{R+G+B}$ $b(R,G,B) = \frac{B}{R+G+B}$
HS	$H(R,G,B) = \arctan\left(\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}\right)$ $S(R,G,B) = 1 - \frac{\min(R,G,B)}{R+G+B}$
$c_1c_2c_3$	$c_1(R,G,B) = \arctan\left(\frac{R}{\max(G,B)}\right)$ $c_2(R,G,B) = \arctan\left(\frac{G}{\max(R,B)}\right)$ $c_3(R,G,B) = \arctan\left(\frac{B}{\max(R,G)}\right)$
$l_1l_2l_3$	$l_1(R,G,B) = \frac{(R-G)^2}{(R-G)^2+(R-B)^2+(G-B)^2}$ $l_2(R,G,B) = \frac{(R-B)^2}{(R-G)^2+(R-B)^2+(G-B)^2}$ $l_3(R,G,B) = \frac{(G-B)^2}{(R-G)^2+(R-B)^2+(G-B)^2}$

3.4.2 Selection of the colour space

In order to obtain a distinctive dominant colour it must be invariant to illumination conditions, shading and viewpoint. These characteristics are highly dependent on the colour space used to represent the radiometric information, as it is shown below. Note also that the fact of selecting the dominant colour makes the descriptor inherently robust to partial occlusion when the physical plane has a clearly defined dominant colour, which is the most common situation. If this is not the case, e.g. a textured plane with different colours, the dominant colour is not a good descriptor and it should not be used for matching.

Different colour spaces have been studied in the context of object recognition in [Gevers and Smeulders, 1999]. This work concludes that normalized *RGB* (*rgb*), saturation and hue (*HS*), and the colour models $c_1c_2c_3$ and $l_1l_2l_3$ (see table 5.1 for the formulation of these colour spaces) are highly invariant to changes in viewing direction and illumination. Below, we analyse these colour spaces for a dataset containing 1000 observations of plane surfaces from different scenarios, spanning diverse viewing conditions (changing viewpoint and illumination, partial occlusion, etc.). As an example, figure 3.4 shows two of these observations, together with the patch histograms in the analysed colour spaces.

Next, we study some relevant properties of such colour spaces:

Histogram invariance. To extract a dominant colour descriptor invariant to viewpoint (including the effects of partial occlusion and shades), the histograms main peak must be stable along different views of the same plane. To measure the histogram stability in a given colour space, we check the similarity of all histograms corresponding to the same plane by means of a chi-squared (χ^2) distance measure [Pele and Werman, 2010]. This measure is used to compute the histogram distances of all pairs of views of the same plane. Then, the mean distance of all analysed pairs is averaged for all tested planes to obtain a global measure of the colour space stability (see table 3.2).

Histogram dispersion. The histograms of planes with a well defined dominant colour must be unimodal and with little dispersion. However, such characteristics do not apply to all the planes in the environment, and also, it varies depending on the colour space. To accept that a plane has a dominant colour we make use of a simple heuristics which requires that at least 50% of the patch pixels are contained in a bandwidth of ±5% of the histogram range, centred at such dominant colour. Thus, we define the concentration rate C as the number of planes that fulfils this condition in all colour components divided by the total number of planes. We have found that the condition above is fulfilled in 97.5% for planes represented with *rgb* and 92.8% for planes represented with $c_1c_2c_3$, while the other colour spaces present much lower rates. Table 3.2 shows the dispersion rate in this experiment, defined as $(1-C)$.

Computation time. Another important criterion to consider is the computation time required to transform the original colour space to the target one. This is less critical because this cost is small in comparison with the whole process of segmenting the planes, whichever the chosen colour space is. The average of this time for this dataset is also indicated in table 3.2.

Table 3.2: Suitability of different colour spaces to represent planar patches according to: histogram stability, histogram dispersion and computation time. The values shown correspond to the average of 100 different planes, with 10 observations each. For all properties, smaller values mean better performance.

	rgb	$c_1c_2c_3$	$l_1l_2l_3$	*HS*
Stability χ^2	0.10	0.11	0.13	0.14
Dispersion $(1-C)$	0.03	0.07	0.74	0.77
Comp. time (μs)	10.7	104.9	23.0	11.3

Taking into account the criteria studied above, we notice that *rgb* is the one with the best properties, and therefore, it is the one adopted here.

3.4.3 Computing the dominant colour

We note that the dominant colour is a discriminative property since its value is repetitive over different observations of the patch (with different viewpoints, lighting conditions, shades and partial occlusions), and it is also distinctive with respect to other patches (different patches have generally different colours). Figure 3.5 shows an example where 5 different planes are observed in different conditions, and still they are easily distinguishable. These observations are represented by their dominant colour, expressed as the histogram mode in each channel in the *rgb* triangle.

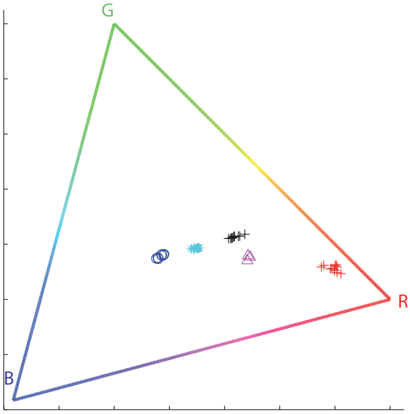


Figure 3.5: Representation of several observations of 5 planar surfaces through their *rgb* mode in the triangular domain of *rgb*. Each planar surface is indicated with a different type of marker.

There exist several ways to define the dominant colour for a planar patch. In this work we have tested the mode of the histograms, and the centroid of the largest cluster extracted with two variants of the mean shift algorithm: with fixed (*FMS*) and variable bandwidth (*VMS*), respectively. Mean shift has been broadly used for colour segmentation [Comaniciu and Meer, 1997]. Though it has limitations for real-time applications due to its computational cost, in our case the cost of the mean shift is affordable since most histograms present unimodal distributions and we only extract one cluster, so that it converges in very few iterations.

We compare the distinctiveness of the dominant colour obtained with the above techniques using a binary classifier based on the colour difference of two patches, expressed as $\|\mathbf{r}_i - \mathbf{r}_j\|$. This difference is actually computed as the L1-norm for each one of the three components in the *rgb* space. Thus, when this difference is larger than a threshold (for any colour component) the patches are considered to belong to different physical planes. This classifier is tested, for a range of thresholds, with the previous dataset in which we know beforehand which observation corresponds to each plane, i.e. we know the classification groundtruth.

From this experiment we obtain the distinctiveness of this classifier in terms of its sensitivity (ratio of actual positives which are correctly identified) and the specificity (ratio of negatives which are correctly rejected) for the different techniques to obtain the dominant colour. These results are depicted as ROC (Receiver Operating Characteristic) curves in figure 3.6. Every point of each curve represents a different threshold for the classifier, thus, more restrictive thresholds result in higher sensitivity

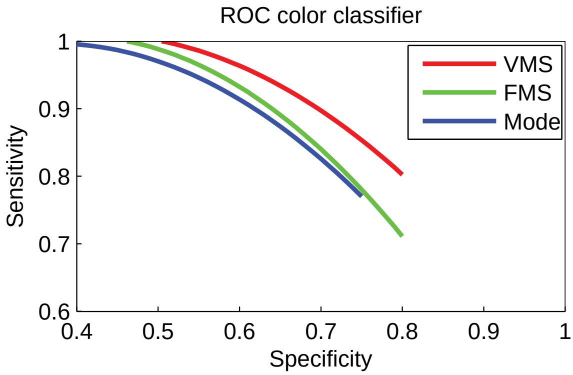


Figure 3.6: ROC curves (sensitivity *vs.* specificity) of the colour constraints as binary classifier.

and lower specificity. Note that the nearer the curve is to the optimum point (1,1) the better the classifier. From this test we conclude that *VMS* provides the most distinctive dominant colour since both, sensitivity and specificity, are higher than for the mode and *FMS* for any threshold.

3.4.4 Choosing a robust colour descriptor

We have seen that the dominant *rgb* colour of a plane is a distinctive property for patch matching in many cases. However, non saturated colours (i.e. $r = g = b = 0.33$), as for instance black and white planes, which are present in many scenarios, cannot be distinguished. Thus, we propose to include in the descriptor the average intensity of the plane so that another loose restriction can be applied to differentiate between such planes. Note that a minimum illumination of the scene is required to use colour in PbMaps, and such a minimum illumination is enough to make a difference between black and white surfaces. This value is calculated as the average $((R+G+B)/3)$ of the inliers supporting the dominant colour given by the previous mean shift segmentation. This parameter permits also recovering the plane's original main colour in *RGB* for visualization purposes.

Also, an important issue when describing patches with their dominant colour is dealing with those cases where this description is not applicable (e.g., textured regions without a prevalent colour). In order to take into account this condition we add a boolean to our colour descriptor to specify whether the distribution of the plane histogram in *rgb* has a low dispersion, as explained in the previous subsection. To sum up, the resulting descriptor contains 4 elements that are stored in a word of 4 bytes: 2 bytes for normalized colour r and g (note that b depends on these two since $r+g+b=1$), 1 byte for the average intensity and 1 byte to specify the existence of a dominant colour.

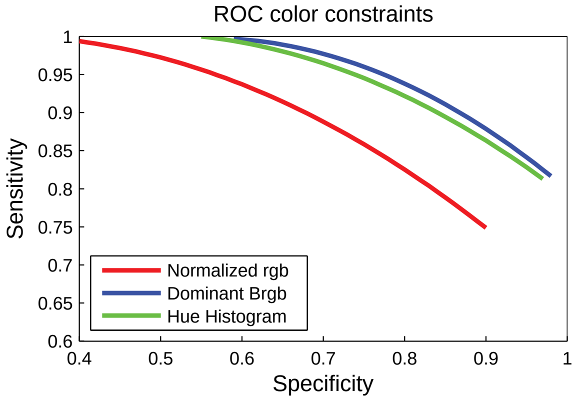


Figure 3.7: ROC curves (sensitivity *vs.* specificity) of different colour descriptors: dominant *rgb*, robust dominant *rgb*++ and hue histogram.

3.4.5 Comparison with the hue histogram

In this section we evaluate the distinctiveness of the proposed colour descriptor by comparing it with the dominant *rgb* colour and with the normalized, saturated hue histogram proposed by [Pathak *et al.*, 2012]. For this last, the histogram distance between $\mathbf{h}_1$ and $\mathbf{h}_2$ is computed with the Bhattacharyya distance [Bhattacharyya, 1946]:

$$B(\mathbf{h}_1, \mathbf{h}_2) = \sqrt{1 - \sum_{k=1}^{N} \sqrt{\mathbf{h}_1[k] \cdot \mathbf{h}_2[k]}} \qquad (3.3)$$

The sensitivity and specificity of a binary classifier based on the compared descriptors are evaluated using different thresholds as we did in the previous section (see figure 3.7). As expected, we observe that the proposed descriptor is significantly more distinctive than the *rgb* dominant colour, since the latter lacks the robust information added to the first. By comparing our descriptor with the hue based histogram we observe that their distinctiveness are similar despite the richer information of the latter. The reason for this is that most planes have a dominant colour in our test environment, as in most indoor scenarios. The fact that the sensitivity of the hue histogram is slightly lower is explained because the histogram is less robust to partial viewing. Contrarily, this descriptor should perform better for textured surfaces and when the patches present no occlusion, however, such cases are rare in the home and office environments we are working in, where our dominant colour descriptor is more suitable.

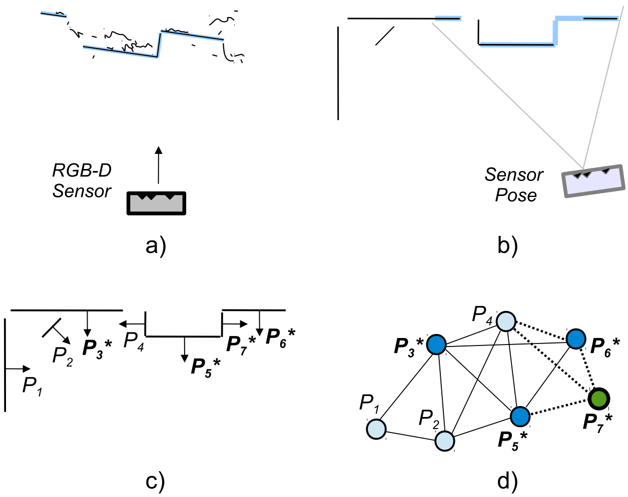


Figure 3.8: 2D representation of the map construction scheme. a) RGB-D capture with segmentated planes (blue). b) Current PbMap with segmented planes (blue) superimposed according to the sensor pose. c) PbMap updated: the planes updated are highlighted d) PbMap graph updated: the planes updated are highlighted in blue, the new plane P_7 is marked in green and, the new edges are represented with dashed lines.

3.5 PbMap construction

After the previous segmentation stage, each detected planar patch is integrated into the PbMap according to the sensor pose, either by updating an already existing plane or by initializing a new one when it is first observed. The sensor pose needed to locate the planes in a common frame of reference can be obtained in different ways. For instance, the current pose may be obtained from the observation of a sufficient number of planes of the PbMap [Fernández-Moral *et al.*, 2013b]. Otherwise, range, visual or combined range and visual odometry may be used [Kerl *et al.*, 2013b; Gokhool *et al.*, 2014].

The PbMap construction procedure is illustrated in figure 3.8. For every new frame, a subsampled point cloud (160 × 120) is built relative to the sensor, and planar patches are segmented from it. The segmented patches are then placed in the PbMap according to the sensor pose (figure 3.8.b). If the new patch overlaps a previous one and their normal vectors coincide, then they are merged and the parameters of the resulting plane are updated. In other case, a new plane is initialized in the PbMap (figure 3.8.c). The graph connections of the observed planes are also updated at every new observation by calculating the minimum distance between the current planes in view and the surrounding planes (figure 3.8.d). An example of a PbMap built from a

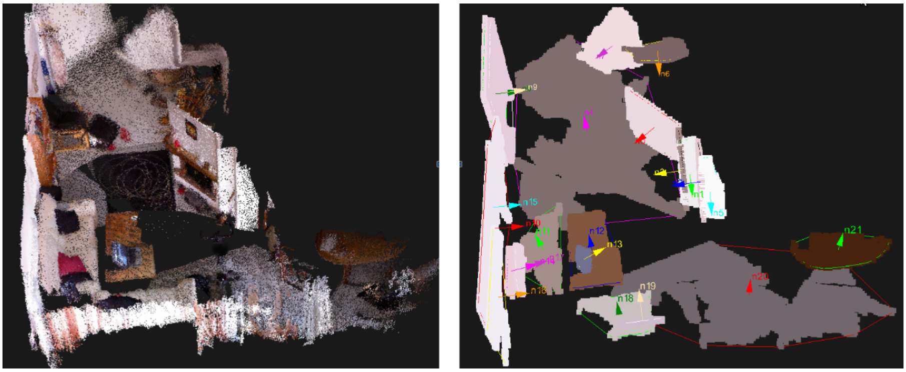


Figure 3.9: Plane based representation of a living room. The coloured planes at the right have been extracted from the point cloud at the left.

short RGB-D video sequence in a home environment is shown in figure 3.9 where we can distinguish the different planes segmented.

3.6 Place recognition and localization in PbMaps

The identification of a place using PbMaps is based on matching and aligning a set of neighbour planes that are represented by a graph. This process can be divided into three different stages: first, the scope and the size of the subgraphs that are to be compared has to be chosen; second, an interpretation tree is applied employing geometric and radiometric constraints to match the maximum number of planes between the two subgraphs; and finally, the matched planes are aligned rigidly, providing an error measurement and the relative rigid transformation between the matched places. These two last stages constitute the technique for scene registration, that can be applied when the first stage is not required as in the registration of PbMaps extracted from single images (i.e. PbMap odometry), or to find the correspondence between PbMaps that have already been localized in the same local region.

3.6.1 Choosing the scope of search

The first question implies that we have to select a set of planes (or subgraph) which defines a place as a distinctive entity. The key to select a subgraph from the multiple combinations that are possible in a PbMap lies in the graph connections, as they link highly related planes in terms of distance and co-visibility. Thus, a subgraph is selected by choosing a reference plane and taking its *k*-order neighbours which are defined by a distance threshold. In the experiments of this chapter we use the first

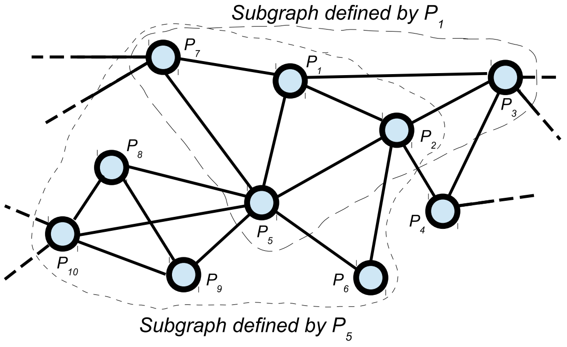


Figure 3.10: Example of the graph representation of a PbMap, where the arcs indicate that two planes are neighbour. Two subgraphs are indicated: the ones generated by the reference planes P_1 and P_5, respectively.

order neighbours and we set a distance threshold of 1 m to define neighbour planes (see figure 3.10). This strategy permits to describe a place in a piecewise continuous fashion, so that different subgraphs can be possible around a local area, providing flexibility to recognize places that are partially observed.

The number of possible subgraphs grows linearly with the map size, that is, the maximum number of subgraphs in the PbMap is limited by the number of planes, though in practice, this number is smaller, since one particular subgraph can be generated from two –or more– neighbour planes (e.g. the subgraphs generated by P_8 and P_9 in figure 3.10 are the same). Also, when a subgraph is contained in other subgraph, only the largest one is considered for matching a place. Thus, in order to achieve a scalable solution for place recognition or loop closure we just need to guarantee bounded time for graph matching.

3.6.2 Graph matching

The problem addressed here is that of matching local neighbourhoods of planes, represented as subgraphs in the PbMap. Thus, we aim to solve a graph matching problem allowing for inexact matching to be robust to occlusions and viewpoint changes. Several alternatives are found in the literature for this problem, from tree search to continuous optimization or spectral methods [Hansen *et al.*, 2008]. Here, we employ a tree search strategy because it does not require further information like the probability of the graph attributes, it is easy to implement and it is extremely fast to apply when the subgraphs to be compared have a limited size. In order to match two subgraphs we rely on an interpretation tree [Grimson, 1990], which employs weak restrictions represented as a set of unary and binary constraints. On the one hand, the unary constraints are used to check the correspondence of two single planes based on the comparison of their geometric and radiometric features. On the other hand, the binary constraints serve to validate that two pairs of connected planes present the same geometric relationship. An important advantage of this strategy is that it allows to recognize places when the planes are partially observed or missing (inexact matching), resulting in high robustness to changes of viewpoint.

3.6.2.1 Unary constraints

The unary constraints presented here are designed to reject incorrect matches of two planes, and thus, to prune the branches of the interpretation tree. Thus, the unary constraints serve to speed-up the search process. These are *weak* constraints, meaning that the uncertainty about the plane parameters is high, so the thresholds are very relaxed to avoid rejecting a correct match. In other words, a unary constraint should validate that two planes are distinct when their geometric or radiometric characteristics are too different, but they lack information to confirm that two observations belong to the same plane, since even different planes can have the same characteristics.

Three unary constraints have been used here, which perform direct comparisons of the plane's area, elongation, and dominant colour if available. For example, the area constraint checks that the ratio between the areas of two observed planes are under certain bounds, and similarly for the other constraints. That is

$$\frac{1}{threshold} < \frac{area_{P_i}}{area_{P_j}} < threshold \tag{3.4}$$

In order to determine appropriate thresholds for such constraints, we analyse their performance in a dataset containing 1000 observations of plane surfaces from different scenarios, spanning diverse viewing conditions (changing viewpoint and illumination, partial occlusion, etc.). We have manually classified these planes, so that the correspondences of all plane observations are known. Then, we analyse the classification results of our constraints in terms of the sensitivity (ratio of actual positives which are correctly identified) and the specificity (ratio of negatives which are correctly rejected), for a set of different thresholds. The result of this experiment are shown with a ROC curve, which shows the sensitivity with respect to the specificity for a given threshold, see figure 3.11. The curves show that higher values of specificity correspond to smaller values of sensitivity and viceversa. Note that the nearer the curve is to the optimum point (1,1) the better the classification of the weak constraint. From this graph we can see that the colour is the most discriminative constraint. Also, since all unary restrictions require similar computation, we arrange them according to their discrimination power, thus the first constraint applied is the radiometric one, followed by the area and the elongation, respectively.

The thresholds for each constraint are determined consistently by choosing a minimum sensitivity of 99%. We notice that those planes that are incorrectly rejected by a unary constraint correspond to planes which have been partially observed (e.g. the

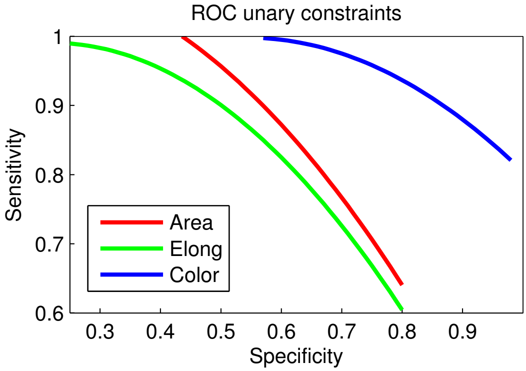


Figure 3.11: Comparison of the different unary constraints by their ROC curves (sensitivity *vs.* specificity).

corner of a table). The fact that some planes might be rejected incorrectly is not critical to recognize a place since not all of the planes are required to be matched. The thresholds obtained here depend on the amount and variety of the training samples used. But since most indoor scenes have planes of similar sizes and with similar configurations, such thresholds must be valid for most similar environments. Besides, we have observed that variations of one order of magnitude in the thresholds do not affect significantly the results for place recognition.

3.6.2.2 Binary constraints

The binary constraints impose geometric restrictions about the relative position of two pairs of neighbour planes (e.g. the angle between the normal vectors of both pairs must be similar, up to a given threshold, to match the planes). These constraints are responsible to provide robustness in our graph matching technique, enforcing the consistency of the matched scene. Three binary constraints are imposed to each pair of planes in a matched subgraph. First, the angle difference between the two pairs being compared should be similar. This is

$$\left|\arccos(\mathbf{n}_i^C \cdot \mathbf{n}_j^C) - \arccos(\mathbf{n}_{ii}^M \cdot \mathbf{n}_{jj}^M)\right| < threshold \tag{3.5}$$

where $\mathbf{n}_i^C$ and $\mathbf{n}_j^C$ are the normal vectors of a pair of nearby planes from the subgraph C, and similarly $\mathbf{n}_{ii}^M$ and $\mathbf{n}_{jj}^M$ are the normal vectors of a pair of planes from the subgraph M.

Also, the distances between the centroids of the pair of planes must be bounded

$$\left|(\mathbf{c}_j^C - \mathbf{c}_i^C) - (\mathbf{c}_{ii}^M - \mathbf{c}_{jj}^M)\right| < threshold \tag{3.6}$$

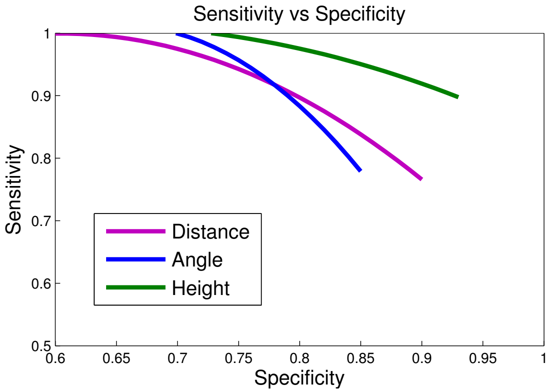


Figure 3.12: Comparison of the different binary constraints by their ROC curves (sensitivity *vs.* specificity).

The other binary constraint takes into account the perpendicular distance from one plane to the centroid of its neighbour. This distance must be similar when the two pair of planes are correctly matched,

$$\left|\mathbf{n}_i^C \cdot (\mathbf{c}_j^C - \mathbf{c}_i^C) - \mathbf{n}_{ii}^M \cdot (\mathbf{c}_{jj}^M - \mathbf{c}_{ii}^M)\right| < threshold \tag{3.7}$$

Other constraints have been tested employing the distance between planes, and the direction of the principal vectors, however, these constraints did not improve significantly the search since they are highly sensitive to partial observation of planes. Nevertheless, these constraints can be used when partial observation is not a big issue, like when matching nearby omnidirectional frames as in chapter 5.

Similarly to the previous subsection, the classification performance of these constraints is analysed for a range of thresholds. For that, we estimate the ROC curves to show the balance between sensitivity and specificity of these binary constraints, which are shown in figure 3.12.

3.6.2.3 Interpretation tree

Algorithm 1 describes the recursive function for matching two subgraphs. This function checks all the possible combinations, defined by the edges among the planes of the subgraphs S_C and S_B, to find the one with the maximum number of matches. In order to assign a new match between a plane from S_C and a plane from S_B the unary constraints are verified first (their result is stored in a look-up table to speed up the search), and if they are satisfied, the binary constraints are checked with the already matched planes. If all the constraints are satisfied, a match between the planes is accepted and the recursive function is called again with the updated arguments. The algorithm finishes when all the possibilities have been explored, returning a list of pairs of corresponding planes.

Algorithm 1 MatchSubgraphs

```
INPUT:
    $S_C$, $L_C$ // Current subgraph and List of planes of $S_C$
    $S_M, L_M$ // Previous subgraph and List of planes of $S_M$
    $matched\_planes$ // List of matched planes
OUTPUT:
    $best\_combination$ // Final list of matched planes

$best\_combination = MatchSubgraphs(LC, LM, matched\_planes)$

best_combination = matched_planes
for each plane $P_C \in L_C$ do
  for each plane $P_M \in L_M$ do
    if $EvalUnaryConstraints(P_C, P_M) == False$ then
      continue
    end if
    for each $P'_C, P'_M \in matched\_planes$ do
      // Check if the edges $P_C, P'_C$ and $P_M, P'_M$ exist
      if $P_C, P'_C \in S_C and P_M, P'_M \in S_M$ then
        if $EvalBinaryConstraints(P_C, P'_C, P_M, P'_M) == False$ then
          continue
        end if
      end if
    end for

    // Remove $P_C$ from $L_C$ and $P_M$ from $L_M$
    $new\_L_C = L_C - P_C$
    $new\_L_M = L_M - P_M$
    $new\_matched\_planes = matched\_planes \cup \{P_C, P_M\}$

    // Search for the best combination of matched planes
    $result = MatchSubgraphs(new\_L_C, new\_L_M, new\_matched\_planes)$

    // Check the length of the resulting list of matched planes
    if $SizeOf(result) > SizeOf(best\_combination)$ then
      $best\_combination = result$
    end if
  end for
end for

return $best\_combination$
```

Despite the large amount of possible combinations for this problem, most of them are rejected in an early stage of the exploration since they do not fulfil the geometric (or radiometric) restrictions. In addition, the evaluation of these restrictions requires little computation, since they only do simple operations to compare 3D vectors and scalars. The cost of this process depends linearly on the number of edges in the subgraphs, and the number of edges has an exponential relation with the threshold defining neighbour planes, and with the number of levels of neighbouring relations used to define the subgraph. This allows the search process to work at frame rate when the number of edges in the subgraphs is bounded (e.g. 10^{25} edges per subgraph, such a number of edges can be obtained by setting a big distance threshold for neighbour planes like $\sim$ 20 m, which is clearly inflated). By considering smaller, more reasonable thresholds to define distinctive contexts of planes, this process performs in the order of microseconds.

Notice that this strategy for place recognition can give rise to several candidate places, one per previous subgraph. A minimum of 4 planes, with a sum of areas bigger than 2 m^2 is required to accept the candidate. And from these candidates, we choose the one with the best rigid alignment, which is given by the consistency test described in the next section.

3.6.3 Localization and rigid consistency

A consistency test is proposed here to evaluate the rigid correspondence of the matched planes of two subgraphs provided by the interpretation tree. This technique requires that at least 3 linearly independent (non parallel) planes are matched to estimate the relative pose between them, μ. This is accomplished by minimizing a cost function which measures the adjustment error of each matched plane. Mathematically

$$\hat{\mu} = \underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{N} e_i(\mu)^2 \tag{3.8}$$

where N is the number of matched planes and $e_i(\mu)$ represents the adjustment error of a pair of planes P_i and P_{m_i} with respect to the rigid transformation defined by μ. This error corresponds to the distance from the centroid of P_i to its matched plane P_{m_i} (refer to figure 3.13). Thus, the proposed error function $e_i(\mu)$ is given by

$$e_i(\mu) = w_i \, \mathbf{n}_{m_i} (\exp(\mu)\mathbf{c}_i - \mathbf{c}_{m_i}) \tag{3.9}$$

being $\mathbf{n}_{m_i}$ the normal vector and $\mathbf{c}_{m_i}$ the centroid of P_{m_i}; $\mathbf{c}_i$ is the centroid of P_i, and $\exp(\mu)$ is the rigid transformation matrix in $\mathbb{SE}(3)$ represented as the exponential map of the 6D vector μ, which is a minimal parametrization for the relative pose, and w_i a weight defined by

$$w_i = \frac{A_i}{\sum_{j=1}^{N} A_j} \tag{3.10}$$

where A_i and A_j are the area of the planes P_i and P_j, respectively. This weight gives more relevance to the adjustment error of larger planes over smaller ones.

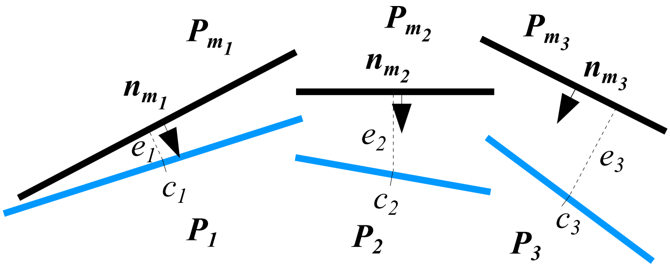


Figure 3.13: Consistency test. 2D representation of the depth error (the blue segments represent planes of the current subgraph and the black segments correspond to a previous subgraph).

We solve this non-linear least squares problem using Gauss-Newton optimization for μ. Notice that other scene alignment methods can be applied, like dense alignment. The election of the one presented here is motivated by two reasons: the residual error couples rotation and translation, and so it constitutes a measure of the quality of the alignment; and second, it is simple and fast to calculate from the information already present in the map.

After the above method has converged and the relative pose has been calculated, the resulting error is used to evaluate the consistency of the candidate matches. In our experiments, we employ the matched area divided by the residual of this optimization to obtain a non-dimensional parameter which represents the quality of the alignment. We have verified empirically that a matched scene with a score higher than 100 almost always corresponds to a correct match (99.8%), therefore, we use this threshold to gain robustness against possible incorrect matches.

3.7 Experimental validation

This section presents the experiments carried out to validate our approach for place recognition. These experiments are divided in two subsections depending on the input data: range only or RGB-D. The difference is relevant since different sensors can be employed for each option. Time-of-flight (ToF) cameras, LIDAR, or structured light cameras like Asus Xtion are valid sources for the case of only range, while the most common option for depth and intensity are RGB-D cameras like Microsoft Kinect or Asus Xtion Pro Live. The advantages of adding radiometric information to the geometric description of a PbMap are also demonstrated here.

In the first set of experiments, the effectiveness for recognizing places is evaluated with 300 tests performed in an environment composed of 15 rooms; second, we evaluate the robustness of our solution to recognize places in non-static scenes, in other words, we evaluate the suitability of the PbMaps to represent scenes that suffer changes continuously (this second experiment is performed using only range images). In these experiments we have employed an Intel Core i7 laptop with 2.2 GHz processor. The experiments are performed using video sequences captured with a RGB-D camera (Microsoft Kinect), where we only use the depth images for the case of range only. In such a way, the results of the two input data can be compared.

3.7.1 Recognition from range images

In the first battery of experiments we explore the scene with a handheld RGB-D sensor, building progressively a PbMap while at the same time, the system searches for previous places. In order to build the PbMap, the pose of each frame is estimated with a method for dense visual odometry[2], which applies the same strategy from [Steinbrücker *et al.*, 2011]. This method estimates the relative pose between two consecutive RGB-D observations by iteratively maximizing the photoconsistency of both images. The optimization is carried out in a coarse-to-fine scheme that improves efficiency and allows coping with larger differences between poses. The drift of this algorithm along the trajectory is sufficiently small to achieve locally accurate PbMaps.

While the scene is explored and the PbMap is built, the current place is continuously searched in a set of 15 previously acquired PbMaps corresponding to different rooms of office and home scenarios (these PbMaps generally capture a 360° coverage of the scene, see figure 3.16). An additional challenge of this experiment comes from the fact that some PbMaps represent the same type of room. This is an important issue for solutions based on bag-of-words since features are normally repeated in scenes of the same kind. In the case of PbMaps, this can also be problematic as some scenes share a similar layout.

We have repeated 20 exploration sequences with different trajectories for each one of the 15 different scenarios. The success and failure rates for place recognition have been recorded, together with the average length of the sensor trajectory until a place was detected, or until the scene was fully observed when no place was recognized. Table 3.3 shows the recognition rate for these experiments. The first column indicates the percentage of cases where a place was recognized correctly, while the failure rate stands for the percentage of places recognized erroneously. The average length of the path taken until a place is recognized is shown in the third column. This somehow gives an idea of how distinctive the local neighbourhoods of planes are for each different scenario. Nevertheless, note that the length of exploration is not directly related to the recognition rate, since even scenes with few distinctive subgraphs (e.g. the case of an empty room) can eventually be matched. An interesting feature of our approach is that it can recognize easily places where there is little appearance information, but where the geometric configuration of planes is highly descriptive, this can be perceived in the video `http://youtu.be/uujqNm_WEIo`. In cases where there are fewer extracted planar patches the recognition rate drops.

A second battery of experiments shows that PbMaps can be used to recognize places that have suffered some changes, but where the main structure of the scene

[2] This method was implemented by Miguel Algaba, and is publicly available at `https://code.google.com/p/photoconsistency-visual-odometry/`

Table 3.3: Effectiveness of the proposed method in different environments with different exploration trajectories (20 tests for each environment). There are some tests where no place was recognized (neither correctly nor erroneously), as a consequence, the sum of the recognition rate and the failure rate is not 100%.

Scenario	*Recog. rate*	*Failure rate*	*Av. path length (m)*
LivingRoom1	100%	0%	5.53
LivingRoom2	100%	0%	3.25
LivingRoom3	100%	0%	2.85
Kitchen1	100%	0%	4.53
Kitchen2	100%	0%	2.24
Kitchen3	90%	0%	3.75
Office1	100%	0%	2.01
Office2	90%	10%	2.61
Office3	90%	10%	3.82
Hall1	100%	0%	1.34
Hall2	80%	10%	2.31
Bedroom1	60%	10%	4.98
Bedroom2	50%	20%	6.25
Bedroom3	55%	20%	5.52
Bathroom	50%	35%	5.60

is unchanged. For that, we have evaluated the recognition rate with respect to the amount of change in the scene, which is measured using Iterative Closest Point (ICP) [Besl and McKay, 1992] on the point clouds built from the depth images. Similarly as in the previous experiments, we evaluate the recognition rate for 20 different trajectories exploring each one of two following scenarios: Office1 and LivingRoom1 (we have chosen these two scenarios because changes are more common in them, see figure 3.14). The results of these experiments are summarized in Table 3.4, showing that the recognition rate remains high for moderate changes in the scene (Ch1 & Ch2, where chairs have been moved, and some objects like a laptop, have disappeared from the scene, while new objects have also appeared), though as expected, this rate decreases as the change in the scene increases significantly (Ch3 & Ch4, where cardboard boxes have been placed in the scene, occluding previous planes and generating new ones).

3.7.2 Recognition from RGB-D images

The same experiment of the previous subsection, in which we explore 15 different scenes with 300 independent sequences is carried out here adding colour information. The main differences are that the main colour of the planar patches is extracted in each observation to update the map, and that the unary colour constraint introduced previously (section 3.4) is added to the interpretation tree for searching previ-

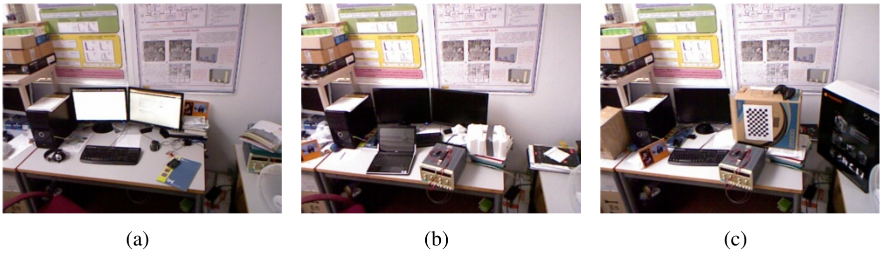


Figure 3.14: Lifelong maps in office environment. a) Reference scene (Ch0), b) Scene with moderate changes (Ch3), c) Scene with significant changes (Ch5).

Table 3.4: Lifelong maps. The ICP fitness score shows the average adjustment error per 3D-point. The recognition shows the percentage of "finds" for 20 different trajectories exploring the scene.

Office1	*Ch0*	*Ch1*	*Ch2*	*Ch3*	*Ch4*
Av. ICP error (mm)	0	0.671	1.215	1.540	3.442
Recognition	100%	100%	95%	90%	80%
LivingRoom1	*Ch0*	*Ch1*	*Ch2*	*Ch3*	*Ch4*
Av. ICP error (mm)	0	1.182	2.010	2.942	3.863
Recognition	100%	100%	100%	95%	85%

ous places. Regarding the PbMap construction, the slow down to compute the main colour is small in comparison with the plane segmentation stage, and the system still works at 30 Hz. Regarding place recognition, or more concretely graph matching, we perceive two relevant improvements: first, the search is more efficient, and second, it is more robust to incorrect matches.

The performance improvement is illustrated with an experiment which shows the number of restrictions checked (which is directly proportional to the time required for searching a place) with respect to the subgraphs size, with and without the use of the colour descriptor. Figure 3.15 shows the average time of the search with respect to the number of planes being evaluated. We observe that performing the search using the proposed colour descriptor is around 6 times faster. Such a rate varies from 2 to 10 depending on the radiometric characteristics of the planar surfaces of the particular environment. This constitutes a significant increase of efficiency over the previous pure-geometric solution.

The radiometric information in PbMaps allows to distinguish different places with similar geometric layout but different colour. That was the case in two bedroom environments of the previous experiment, where colour information helps to differentiate one from another. The results show that apart of the improvement on efficiency, the solution employing colour is more robust to incorrect matching. This is shown in table 3.5, where a significant reduction in the number of mismatched scenes is achieved.

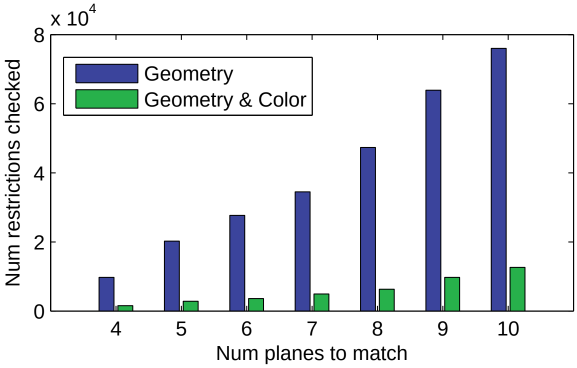


Figure 3.15: Performance of the place recognition process (in terms of the number of restrictions checked until matching with respect to the size of the subgraph to match) for both: only geometry and colour and geometry in PbMaps. The computing time is directly proportional to the number of restrictions checked.

Table 3.5: Robustness to wrong recognition by using colour information.

Scenario	*Failure rate (depth)*	*Failure rate (depth+colour)*
Office2	10%	0%
Office3	10%	0%
Hall2	10%	0%
Bedroom1	10%	5%
Bedroom2	20%	10%
Bedroom3	20%	5%
Bathroom	35%	30%

3.8 Discussion

This chapter presents a highly compact map representation of the scene based on planar patches (PbMap). Such planes are efficiently extracted from range images with a region growing procedure, permitting the use of this representation for real-time mapping and SLAM. The planes are described by simple geometric attributes, and also colour information if it is available. A PbMap is structured as an undirected graph where the nodes represent planes and the edges store the neighbour relations between them, so that they contain contextual information. This arrangement permits to operate quickly on local neighbourhood of planes for scene registration and place recognition.

A new methodology for real-time place recognition in indoor environments using PbMaps has been proposed. The recognition process is tackled with an interpretation tree, which matches efficiently local neighbourhoods of planes based on weak constraints that prune the match space. This matching process employs unary and binary constraints. The unary constraints restrict the individual correspondence of pairs of matched planes, and its main contribution is the speed-up of our solution. On the other hand, the binary constraints check that the layout of the scenes being compared are geometrically consistent, and so, they are responsible of the robustness of this technique.

This kind of map is interesting for representing indoor scenes, where the amount of planar surfaces dominates over non-planar structures. The proposed solution can work with range cameras, by generating a geometric description of the scene, or with RGB-D sensors, adding radiometric information to the planes to improve the description and the recognition performance through the unary constraints. A colour descriptor consisting of the dominant colour, the average intensity and a parameter indicating the robustness of dominant colour was employed. A comparison of both alternatives (only geometry *vs.* geometry and colour) has been presented, which shows an average speed-up of 6 times in scene recognition by using colour information.

A registration technique is proposed to further check the metric consistency of two matched places, and to recover the metric localization. This technique aligns the scenes through the minimization of a cost function, whose residual is used to validate the proposed match. This minimization provides the relative pose between the matched places, which can be used for loop closure for instance. We provide experimental results demonstrating the effectiveness of our approach for recognizing and localizing places in a dataset composed of 20 home and work-place scenes: offices, living rooms, kitchens, bathrooms, bedrooms and corridors.

Apart from the above mentioned advantages, this strategy to describe and identify places is robust to changes in the scene. The point is that most of the movable objects in indoor environments are not planar, and regarding the planar structure, the larger (dominant) planes are generally static. Thus, the approach presented is conceptually adapted to lifelong mapping. In order to test this idea, we performed an experiment to measure how the recognition performance is affected by the fact that objects can be moved by the users. The results confirm the intuition, though a deeper study must be carried out to evaluate the applicability of our representation for such a problem. This constitutes a field for future research after this thesis.

A future improvement for the PbMap will consist of its integration into a probabilistic framework to represent the plane parameters. This will permit to deal with sensor noise to obtain more robust and accurate PbMap registration. Another important improvement would be the ability to detect planar patches that fully cover the represented surface. This implies that the surface is seen with no occlusions, so that information about other dimensions can be exploited to improve localization and to address related tasks like object recognition. Also, an interesting open issue from this research is how to use the compact description of a PbMap for semantic inference, which can provide extra capabilities in mobile robotics and better communication interfaces human-robot. Since the PbMap's compact geometric (and radiometric) description is useful to match scenes, it is reasonable that they can be useful to identify classes of scenes (e.g. kitchens, bedrooms, etc.) what is interesting for example for domestic service robotics. However, this problem has a total different perspective, and a whole research to find distinctive cues must be carried out before evaluating the potential capabilities of PbMaps for this problem.

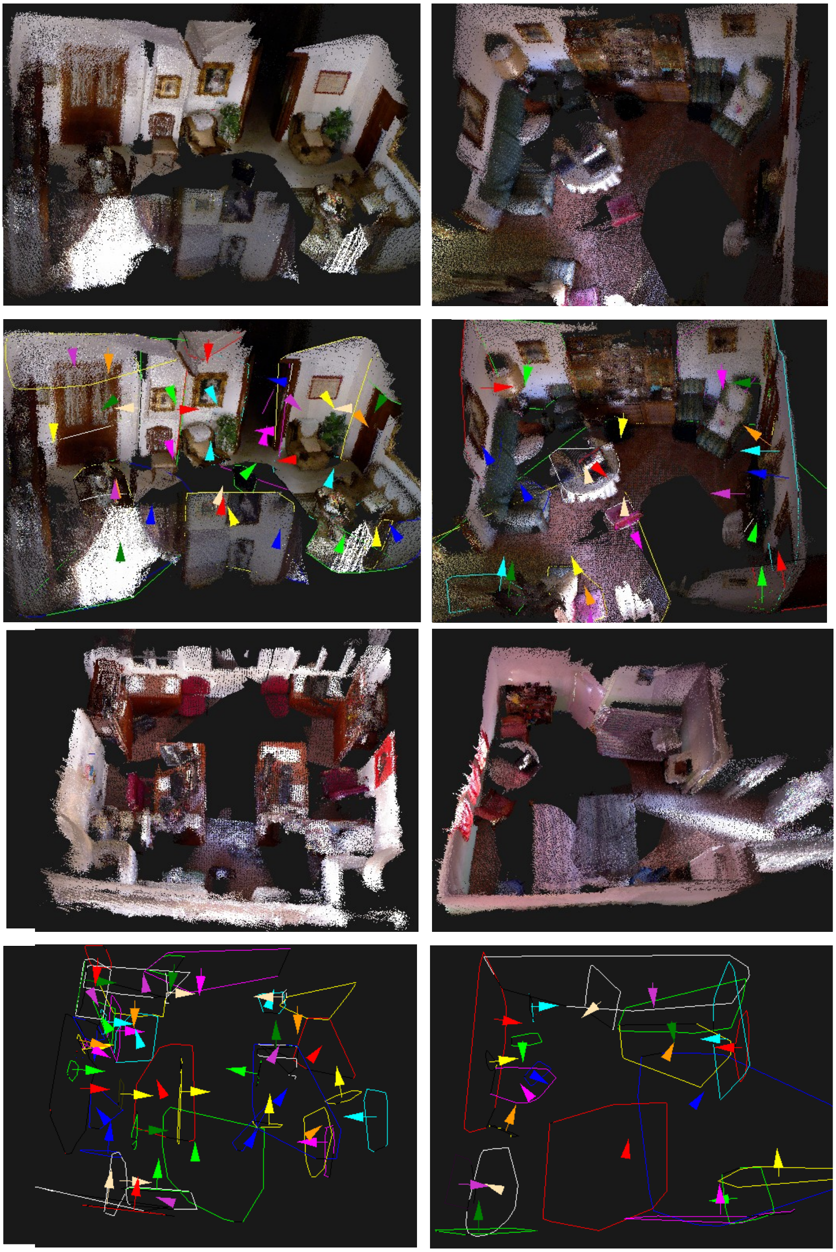


Figure 3.16: Different scenarios where place recognition has been tested. These pictures show the point clouds of some of the maps created previously, showing their PbMap right below of each scenario.

Chapter 4
Hybrid metric-topological mapping

Abstract

Efficient map representations are required in mobile robotics to perform complex tasks. The integration of metric and topological information has been proposed to create multi-layer hybrid maps, where the metric layer is generally used for accurate localization in a local environment, and the topological layer stores high level symbolic information which may be required for planning and task reasoning. This chapter presents a new methodology to structure dynamically a metric map into a topological arrangement of local maps, while the topological structure keeps information about the connectivity of these local maps. The division is carried out based on the co-visibility of map features, and is executed with graph cut. This strategy permits scalable localization and mapping approaches by considering only the set of local maps which are closer to the robot. Experimental results are presented with a monocular SLAM system in indoor and outdoor scenarios.

4.1 Introduction

Different kinds of mapping strategies are needed in mobile robotics to perform different actions. For example, grasping an object requires a metric map of the object and its environment, while for navigation, besides the local metric map required to execute immediate movements, a global map with topological information is generally required to decide the most appropriate path or to reason about the aimed tasks. Such topological representation could encode information that is meaningful also for humans, like the connectivity of rooms in a building. Hybrid metric-topological maps have been proposed for dealing with these two types of information [Thrun, 1998]. In this chapter we build up on previous works to present a dynamic arrangement of the metric-topological representation according to the sensed space [Blanco *et al.*, 2006].

The problem of scalable SLAM is present when coping with some real autonomous robotics applications. Such ability to operate in large scale brings the need of appropriate strategies for managing the map. This problem may be addressed using metric-topological or multilayer hierarchical representations. In this sense, applying abstraction (as humans do) is an effective way of dealing with the huge amount of detail present in large metric maps. The result of such abstraction process is a metric-topological map, consisting of a two-layer representation, one containing pure geometrical information and a second one containing higher level symbolic information [Thrun, 1998]. Thus, the benefit of a metric-topological arrangement is twofold: first, it offers a natural integration with symbolic planning that permits a robot to reason about the world and to execute high level tasks [Galindo *et al.*, 2008]; second, the efficiency and scalability of SLAM is improved by limiting the scope of localization and mapping to the region of the environment where the robot is operating. Also, loop closure and re-localization can be more efficiently solved using topological information [Savelli and Kuipers, 2004; Angeli *et al.*, 2009].

Here, we present a mapping strategy where the metric map is dynamically divided into regions (submaps) with highly connected observations, resulting in a topological structure where each node stores a local metric map, and the arcs represent the relations with neighbouring submaps. The key idea is to cluster in the same submap those features which are more interrelated according to the sensor's visibility, that is, grouping co-visible features, or keyframes with higher overlap depending on the metric map chosen. The map division is performed using a graph cut technique which can be executed online (with every new observation), allowing efficient and scalable SLAM operation.

The mapping approach presented here is tested in the framework of a monocular SLAM, where the experiments show that this hybrid metric-topological approach outperforms the efficiency and scalability of the pure metric approach. Concretely, we will focus on the benefits of such hybrid mapping applied to a well known monocular SLAM system based on Bundle Adjustment on keyframes (PTAM [Klein and Murray, 2007]). Our approach has also been applied in a SLAM solution based on PbMaps [Fernández-Moral *et al.*, 2013b] which is presented in the next chapter.

4.1.1 Related works

Hybrid mapping and map partitioning

Hybrid maps that combine metric and topological information have been proposed for SLAM in large and complex robot environments. Such maps are usually composed of local metric maps (suitable for robot localization) organized in a topological graph structure, which stores the relations between local maps, and/or other high level symbolic information [Bosse *et al.*, 2003; Estrada *et al.*, 2005; Blanco *et al.*, 2009a]. A key question for such hybrid mapping is how the map should be partitioned into local maps.

Map division has been addressed in a number of works. Some relevant examples are: the Atlas framework [Bosse *et al.*, 2003], where a new local map is started whenever localization performs poorly in the current local map, or the hierarchical SLAM presented in [Estrada *et al.*, 2005], where sensed features are integrated into the current local map until a given number of them is reached. However, none of these provides a mathematically grounded solution based on the particular perception of the scene.

In [Eade and Drummond, 2007], the map is divided in nodes where the landmarks are represented in a local coordinate frame and, these landmarks are updated using an information filter. This method uses the common features between adjacent nodes to calculate their relative pose. A different approach called Tectonic-SAM [Ni *et al.*, 2007] uses a "divide and conquer" approach with locally optimized submaps in a Smoothing and Mapping framework (SAM). This approach is improved in [Ni and Dellaert, 2010] to build a hierarchy of multiple-level submaps using nested dissection.

Other works employ "graph cut" to divide the map according to a measurable property of the map observations. On that mathematically sound basis, [Zivkovic *et al.*, 2005] addresses the problem of automatic construction of a hierarchical map from images; [Blanco *et al.*, 2008] generates metric-topological maps using a range scanner, and generalizes the approach for other sensors; and [Rogers and Christensen, 2009] splits the map within a Bayesian monocular SLAM framework to reduce the problem complexity.

Our method, which also relies on graph cut, differs from the works above in the way the graph is updated, which is specifically tailored for online SLAM operation. Our approach resembles also the stereo-SLAM framework of [Lim *et al.*, 2011] who divide the map keyframes into groups (called segments) according to their geodesic distances in the graph. On the contrary, our map partitioning is independent of the keyframe positions, and is only based on observations acquired from the scene. Concretely, the map is split where there are fewer shared observations, minimizing the loss of information and therefore, enforcing the coherency and consistency of the submaps.

Monocular SLAM

Many solutions have been presented to build metric maps with monocular SLAM since [Davison, 2003] presented the first real-time solution for the problem in 2003. Two main strategies have been applied since then: Bayesian filtering (following the work of Davison) and Bundle Adjustment (BA) on keyframes, as introduced in [Klein and Murray, 2007]. The latter represents the base for the current state-of-the-art since it allows handling denser maps and generally offers a better ratio accuracy/cost [Strasdat *et al.*, 2010].

BA, traditionally used as an offline method for Structure from Motion (SfM), is now widely used in visual SLAM thanks to the introduction of parallel processing and efficient algorithms which exploit the sparse structure of the problem. Its application to visual SLAM was inspired by real time visual odometry and tracking [Nistér *et al.*, 2004], where the most recent camera poses where optimized to achieve accurate localization. In a similar vein, PTAM selects keyframes and applies BA in a fixed size window, around the last keyframe incorporated, to optimize the metric map and the camera trajectory. Then, once the local optimization is performed, a low priority global BA is run to improve the map consistency. This approach is extended in [Holmes *et al.*, 2009] by combining it with relative bundle adjustment (RBA) [Sibley *et al.*, 2009], allowing fixed-time, consistent exploration. An improvement of the latter to exploit the problem' sparse structure was recently presented by [Blanco *et al.*, 2013].

The work of [Strasdat *et al.*, 2011] which is also related to RBA, proposes a double window optimization: a first window as in PTAM and a second one including the periphery of the first to improve consistency by optimizing a pose-graph of keyframes. Despite the impressive results obtained, such unique map solution has intrinsic limitations for managing maps of real large environments. To prevent this limitation, we propose a topological arrangement of the map into local metric maps.

4.1.2 Contribution

In this chapter we present a mapping strategy where the metric map is dynamically divided into regions with highly connected observations, resulting in a topological structure which permits the efficient augmentation and optimization of the map. With such map division, the current submap always contains the most relevant metric information about the current robot location, which is useful to improve the efficiency of SLAM. This hybrid mapping solution has been integrated with a monocular SLAM system to demonstrate the advantages of our approach. This strategy can be applied to other types of SLAM as it is demonstrated in the next chapter, where it is applied to an omnidirectional RGB-D SLAM approach.

Subsequently, we describe our metric-topological mapping approach and the map partitioning procedure (section 4.2) and show how it is combined with the monocular SLAM system of PTAM (section 4.3). The experiments and their results are presented next (section 4.4), and finally, we expose our conclusions from these results.

4.2 Hybrid metric-topological mapping approach

Splitting a map into locally consistent metric representations and globally coherent regions provides some relevant advantages for SLAM. Next, we explain the benefits of such map structure (subsection 4.2.1), and describe our proposal to obtain such metric-topological arrangement of the map (subsection 4.2.2). In order to be consistent with the experiments in this chapter, the next subsections tackle specifically SLAM based on bundle adjustment, and for a mapping approach based on keyframes and point features. However, both can be easily generalized to be applied to other types of SLAM approaches, like e.g. pose-graph SLAM or EKF-SLAM, and for other kinds of maps, e.g. PbMaps.

4.2.1 SLAM Improvements through hybrid mapping

The advantages of applying a coherent map partition in SLAM are diverse: a) all the metric data in each submap (which may include the keyframe poses, landmark positions, point clouds, etc.) can be referred to a local coordinate system to reduce error accumulation and to avoid numerical instability; b) localization can be achieved more efficiently since only those map features in the nearer regions are used to estimate the pose of the robot; and c) this map structure permits to approximate the global map optimization by the individual optimization of the different submaps, thus reducing the computational cost of this process. This last advantage is of special relevance due to the demanding nature of map optimization. For bundle adjustment, its complexity ranges from linear to cubic in the number of keyframes depending on the particular structure of the problem [Konolige, 2010]. Let us now explain the details of this approximation for BA global optimization.

Having a map of n landmarks obtained from m keyframes, bundle adjustment can be expressed as

$$\min_{\mathrm{T}_j,\mathbf{p}_i} \sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij}\, d(\mathbf{Q}(\mathrm{T}_j, \mathbf{p}_i), \mathbf{x}_{ij})^2 \tag{4.1}$$

where

- $d(\mathbf{x},\mathbf{x}')$ denotes the Euclidean distance between the image points represented by vectors $\mathbf{x}$ and $\mathbf{x}'$,
- T_j is the pose of camera at keyframe j and $\mathbf{p}_i$ the position of landmark i,
- $\mathbf{Q}(\mathrm{T}_j, \mathbf{p}_i)$ is the predicted projection of landmark i on the image associated to keyframe j,
- $\mathbf{x}_{ij}$ represents the observation of the i-th 3D landmark by keyframe j, and
- v_{ij} stands for a binary variable that equals 1 if landmark i is visible in keyframe j and 0 otherwise.

Let's now consider that the map is divided into N submaps, each submap, say k, containing m^k keyframes and n^k landmarks, with $k = \{1, \ldots, N\}$. Then, (4.1) can be rewritten as

$$\min_{\mathrm{T}_j^l, \mathbf{p}_i^l} \sum_{k=1}^{N} \sum_{l=1}^{N} \left(\sum_{i=1}^{n^k} \sum_{j=1}^{m^l} v_{ij}^{kl} \, d(\mathbf{Q}(\mathrm{T}_j^l, \mathbf{p}_i^k), \, \mathbf{x}_{ij}^{kl})^2 \right) \tag{4.2}$$

where the combination of subscript i and superscript k refers to the i-th landmark of the k-th submap (e.g., $\mathbf{p}_i^k$), and similarly l over j refers to the j-th keyframe of the l-th submap (e.g., T_j^l). Taking into account the observations shared between submaps, this expression can be written as

$$\min_{\mathrm{T}_j^l, \mathbf{p}_i^k} \sum_{k=1}^{N} \left(\underbrace{\sum_{\substack{l=1 \\ l \neq k}}^{N} \sum_{i=1}^{n^k} \sum_{j=1}^{m^l} v_{ij}^{kl} \, d(\mathbf{Q}(\mathrm{T}_j^l, \, \mathbf{p}_i^k), \, \mathbf{x}_{ij}^{kl})^2}_{A} + \underbrace{\sum_{i=1}^{n^k} \sum_{j=1}^{m^k} v_{ij}^{kk} \, d(\mathbf{Q}(\mathrm{T}_j^k, \, \mathbf{p}_i^k), \, \mathbf{x}_{ij}^{kk})^2}_{B} \right) \tag{4.3}$$

where the term A stands for the reprojection error of those landmarks observed from keyframes of different submaps and the term B corresponds to the reprojection error of those landmarks observed from keyframes within the same submap.

If we are able to divide the map in such a way that the different submaps have few common observations, and assuming that the reprojection errors are independent of the map division, then A becomes negligible with respect to B. Thus, the global optimization can be approximated by

$$\sum_{k=1}^{N} \left(\min_{\mathrm{T}_j^k, \mathbf{p}_i^k} \sum_{i=1}^{n^k} \sum_{j=1}^{m^k} v_{ij} \, d(\mathbf{Q}(\mathrm{T}_j, \, \mathbf{p}_i), \, \mathbf{x}_{ij})^2 \right) \tag{4.4}$$

As stated previously, the reduction in complexity is a direct consequence of using submaps. Such a reduction comes from the approximation of the full map optimization by the optimization of each submap independently, which leads to a significant reduction of computational burden. In fact, this approximation is equivalent to the original expression (4.1) when there are no connections between the submaps.

4.2.2 Map partitioning method

The approach proposed here to divide the map into coherent regions consists of grouping together those keyframes that observe the same features from the environment. To that effect, we consider the map as a graph whose nodes represent keyframes and the weight of the arcs are a measure of the common observations between them. There are two critical issues in this partitioning approach: first, the computation of the arc weights; and second, the criterion adopted to perform the partition itself. As for the first, the arc weights are assigned according to the Sensed-Space-Overlap (SSO) following the work of [Blanco *et al.*, 2006], which is particularized here for landmark observations. This simple but effective measure represents the information shared

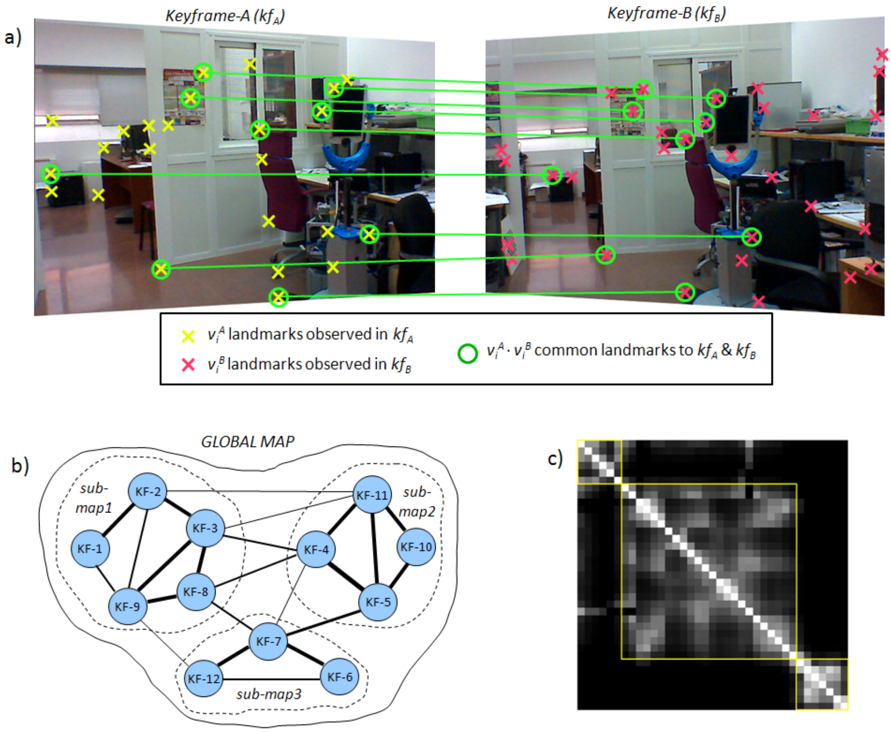


Figure 4.1: a) Common observations between two keyframes. This is used to calculate the Sensed Space Overlap (SSO) (see equation 4.5). b) Graph-representation of the map where each node represents a keyframe and the arcs are weighed with the SSO calculated between keyframes (thicker arcs mean higher SSO). c) Example of SSO matrix, in which the brightness of the element ij represents the SSO between the keyframes i and j.

by two keyframes. It is calculated as the relation between the number of common landmark observations divided by the total number of landmarks observed in both keyframes (see figure 4.1.a). This is expressed as

$$SSO\left(kf_A, kf_B\right) = \frac{\sum v_i^A \cdot v_i^B}{\sum v_i^A + \sum v_i^B - \sum v_i^A \cdot v_i^B} \tag{4.5}$$

where v_i^A and v_i^B, similarly to the definitions of the previous section, are binary variables that equal 1 if landmark i is observed in the keyframes kf_A and kf_B, respectively. We assume that the ratio between outliers and inliers is very low, so that the computed SSO is very accurate. This situation applies to the monocular SLAM system employed in our experiments and to other registration frameworks in SLAM like the PbMap registration, which is employed in the next chapter.

Regarding the criterion for partitioning the graph, we follow previous works [Zivkovic *et al.*, 2005; Blanco *et al.*, 2008; Rogers and Christensen, 2009] that apply the minimum normalized-cut (min-Ncut), originally introduced by [Shi and Malik, 2000]. The min-Ncut has the desirable property of generating balanced clusters of highly interconnected nodes, in this case clusters of keyframes that cover the same part of the environment. Figure 4.1 illustrates this concept, where figure 4.1.a shows the common observations in a pair of keyframes whose arc is calculated through the SSO (eq. 4.5), and figure 4.1.b shows a map division into three submaps as a result of applying min-Ncut. Notice that the pairs of keyframes with higher SSO (thicker arcs) are grouped together. Figure 4.1.c shows the symmetrical SSO matrix corresponding to a different, larger map, where the keyframes are arranged according the min-Ncut to give rise to three groups of keyframes or submaps (matrix blocks).

It is important to notice that, in order to guarantee a scalable system when applying map partitioning in SLAM, the size of the submaps (i.e. number of keyframes) must be kept bounded. This requirement is not demonstrated mathematically here, but it is intuitive to see that as the camera explores new parts of the scene, the new keyframes will have low SSO values (if any) with distant ones in the map. Therefore, the min-NCut will produce new partitions when the system explores unobserved regions of the environment. This can be more clearly understood with the following example: let's consider the case where there are features that are always observed (e.g. the horizon when travelling by train, or when zooming in the scene, or traversing a corridor with the camera pointing in the movement direction) as new keyframes are selected, they will introduce new features and therefore they will contribute to reduce the minimum normalized-cut, resulting in the eventual partition of the map. The last two examples represent another advantage of our partition method, which produces natural multi-scale maps when the camera zooms. This insight is supported by all the experiments we have carried out during this work.

4.3 Metric-topological monocular SLAM

This section outlines the combination of our partition procedure and Parallel Tracking and Mapping (PTAM) [Klein and Murray, 2007]. PTAM is a monocular SLAM algorithm which performs online BA on keyframes, separating the tracking and mapping stages in two different threads to permit efficient real-time performance. This technique requires an initial map before it starts working automatically. Such initial map is acquired with a Structure from Motion procedure that involves user intervention to select two views with sufficient parallax. Once the initial map has been created, the system analyses the images retrieved by the camera to self-localize in the map, while the map is continuously optimized and augmented with new keyframes and landmarks. Such keyframes are selected according to some simple heuristics (see [Klein and Murray, 2007] for more details), and new landmarks are extracted by matching point features between each new keyframe and its nearest keyframe in the map applying epipolar restrictions.

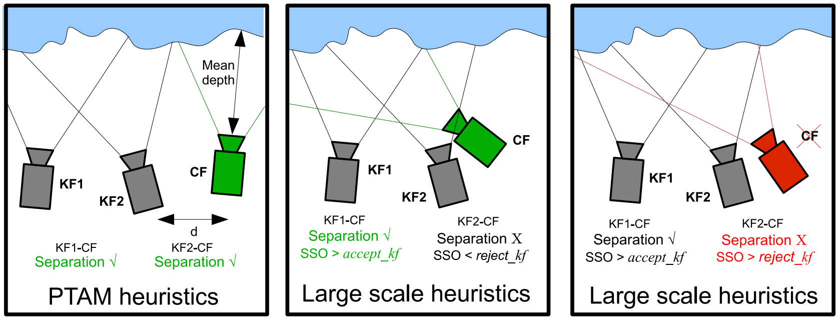


Figure 4.2: Keyframe selection heuristics. a) PTAM's separation condition. b) and c) Keyframe acceptance and rejection heuristics, respectively, for large scale mapping. The thresholds used in our experiments for $accept\_kf$ and $reject\_kf$ are 0.2 and 0.7 respectively.

4.3.1 Keyframes selection in large environments

The keyframe selection criterion becomes an important aspect when PTAM is employed to build maps of large spaces. PTAM was designed for small environments (e.g. an office), where it works adequately with a hand-held camera which is waved sideways. PTAM employs a heuristic rule to select a new keyframe when there is a minimum separation between the current frame and the nearest keyframe in the map (i.e. Euclidean distance divided by the mean depth of the scene). This condition selects valid keyframes when the camera is moved sideways. But unlike in PTAM, we wish to explore big scenes and to construct large maps without being restricted to move sideways. Therefore, we have adapted this heuristic to select a keyframe when it provides useful information for mapping, by adding two more restrictions to the previous one for camera movement. Consequently, the current frame (*CF*) is selected as a new keyframe when:

- There exists a nearby keyframe which meets PTAM's separation condition with *CF* and which shares some information about the scene $(SSO > accept\_kf)$.
- There is not a nearby keyframe which does not meet PTAM's separation condition with *CF* and which shares much information about the scene $(SSO > reject\_kf)$.

Figure 4.2 shows the adapted heuristics to select keyframes in large scale. PTAM's separation condition is shown in figure 4.2.a, where a keyframe is accepted when the Euclidean distance to the nearest keyframe divided by the mean depth of the scene is over some defined threshold. Figure 4.2.b shows the new acceptance condition, which selects the current frame if there exist at least one keyframe that fulfills PTAM's separation and whose $SSO > accept\_kf$ (KF1-CF). Figure 4.2.c shows the rejection condition, which rejects the current frame if there exist at least one keyframe that does not fulfil PTAM's separation and whose $SSO > reject\_kf$ (KF2-CF). These thresholds are selected heuristically after a few quick tests looking for robust localization with a minimum number of keyframes using a hand-held camera. We observe that the values $accept\_kf = 0.2$ and $reject\_kf = 0.7$ produce good results in such sense. This matter is not further investigated here since it is out of the scope of this research.

So, the acceptance condition prevents taking a new keyframe which shares little or no information with the map, while the rejection condition avoids selecting keyframes that are too similar to those already in the map. Hence, the combination of these two conditions permits selecting keyframes that provide new information to the map relaxing the movement constraints for nimble exploration of the scene.

Algorithm 2 Map Partitioning

M and *KF* are a submap and a keyframe respectively. *SSO_M* is the matrix containing the SSO values between all pairs of keyframes in the neighbourhood V. The *current_map* is the submap being tracked. *num_KF* is a keyframes counter and *N_part* is a parameter to control when the partition is to be reevaluated. A keyframe's *match_map* is the submap where it will be added, and a keyframe's *match_KF* is the keyframe used to find point correspondences.

After new keyframe *new_KF* is selected

```
num_KF ++
Select match_map and match_KF
if match_map != current_map then
  num_KF = 0
end
Extract new map-points
Add a new row and a new column to SSO_M
for all submaps $M_i \in V$ do
  for all keyframes $KF_j$ of $M_i$ do
    $SSO\_M \leftarrow SSO(new\_KF, KF_j)$
  end
end
if (num_KF % N_part) == 0 then
  Evaluate partition
  if partition is modified then
    Lock tracking thread
    for all submaps $M_i \in V$ do
      Restructure $M_i$
    end
    Unlock tracking thread
    Update SSO_M
  end
end
```

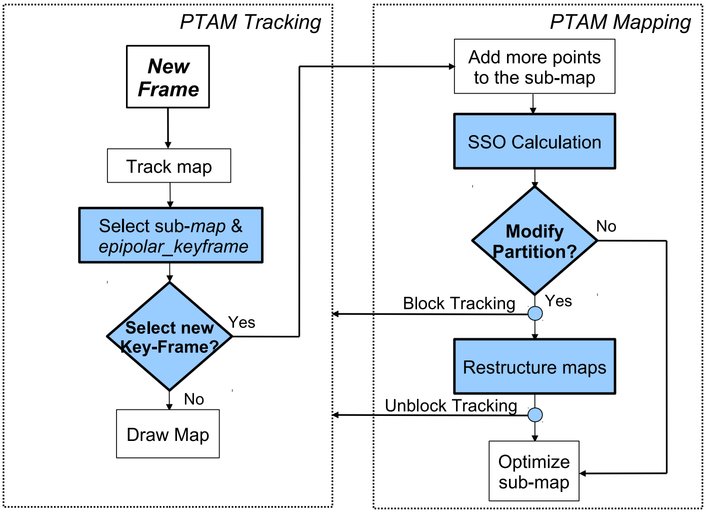


Figure 4.3: Tracking and mapping threads of PTAM. Blue boxes correspond to the embedded stages to perform the map partitioning.

4.3.2 Combination of map partitioning and PTAM

A scheme of the proposed partitioning method interacting with PTAM is depicted in figure 4.3. Our submapping procedure takes action in both of PTAM threads. In the tracking thread, it selects the current submap and the nearest keyframe to the estimated pose after a new frame is analysed. In the mapping thread, after a new keyframe is selected and new landmarks are detected in it, the SSO is evaluated with respect to all the keyframes of the neighbourhood. Such neighbourhood includes all the submaps directly connected to the current submap (see figure 4.4).

The partitioning procedure comes into play after the SSO has been updated, then, the min-Ncut is evaluated, and if it results in a different partition, the map is rearranged. This procedure is described in algorithm 2. This partitioning method is applied dynamically while the map is built and may create new submaps as well as merge existing submaps to maintain the division coherency by grouping keyframes with high overlap. The result is a metric-topological map, where two different topological areas will be connected by a rigid transformation if there are common observations between them.

The partitioning process, including SSO computation, min-NCut evaluation and map rearrangement depends on the number of keyframes and landmarks in the neighbourhood, taking up to 100 ms in our experiments, which is a short time in comparison with the map optimization stage through BA.

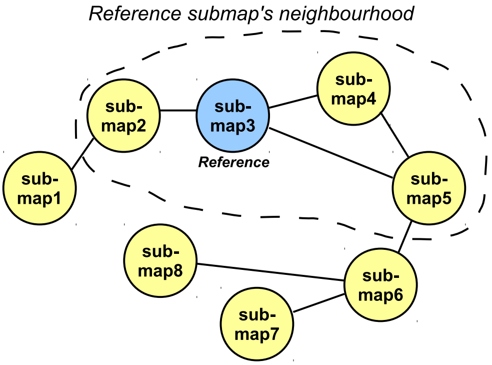


Figure 4.4: Topological representation of the map, showing the neighbourhood of a reference submap.

Figure 4.5: Experimental set up: laptop with attached camera.

4.4 Experiments

In this section we present some experiments which show the advantages, in terms of efficiency and scalability, of using the proposed metric-topological arrangement of the map instead of a single metric map. The experiments have been carried out using a Philips SPC640NC webcam, connected by USB to a linux-based laptop with an Intel Core2 Duo 2.4 GHz processor, 2Gb of memory and a nVidia GeForce-9400 graphics card. The camera intrinsic parameters were calibrated using the methodology described in section 2.2.1. Figure 4.5 shows the set up of our monocular SLAM system.

A first experiment is aimed to illustrate the increase of efficiency in localization (tracking of the camera's pose). For that, we compare the time needed to reproject

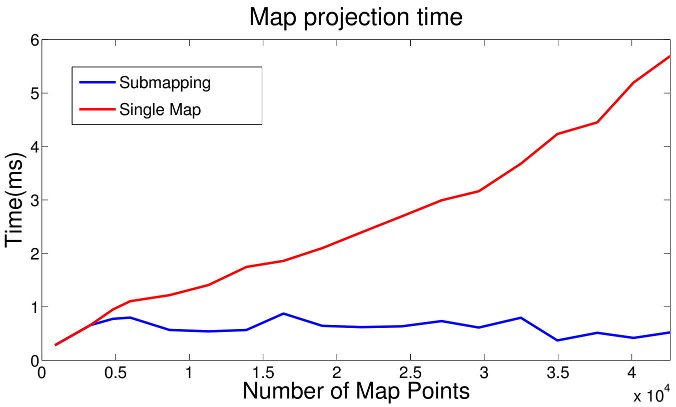


Figure 4.6: Map projection time for localization with and without map partitioning.

map points into the current frame with and without partitioning as the map grows. Both tests have been performed in the same environment, building maps composed of about 45K points and 1K keyframes, distributed in 52 submaps for the partitioning case. Figure 4.6 shows that the time with a unique map grows linearly with the number of map points, whereas with the metric-topological submapping this time is bounded since only those points in local maps close to the camera are evaluated. This improvement in efficiency becomes more evident when the map grows non-stop (note that this process is performed with each new frame captured by the camera, at 30 Hz).

The goal of the second experiment is to quantify the efficiency in the global optimization of the map with our submapping approximation. For that, we have run BA offline after every new keyframe is selected from a recorded video (that is, sequential SfM), measuring the times of each BA completion with and without partitioning. At the end of these tests, the maps created were composed of about 22K points and 400 keyframes, distributed in 9 submaps for the partitioning case. In order to compare both alternatives in the same conditions, we have included the time of partition management in the BA time for the partitioning test. Figure 4.7 shows the computing time of the optimization *vs.* the number of keyframes of the whole map for both cases. As expected, for the case of a single metric map, the computational cost follows an increasing polynomial trend with the number of keyframes. Conversely, when applying hybrid mapping, the computational burden is bounded since the BA is applied only on the current submap. For this case, we can observe some abrupt changes in the cost which are produced when the reference submap (the one where the system is localized) switches to a neighbour of different size. Figures 4.8.a and 4.8.b show the maps built with both alternatives (different colors represent different submaps in 4.8.b). We can verify visually their high similarity, and their good alignment, as a result of the continuous optimization previous to the map partition.

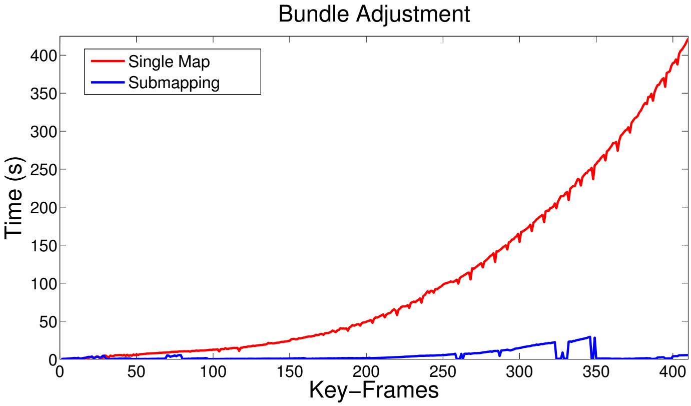


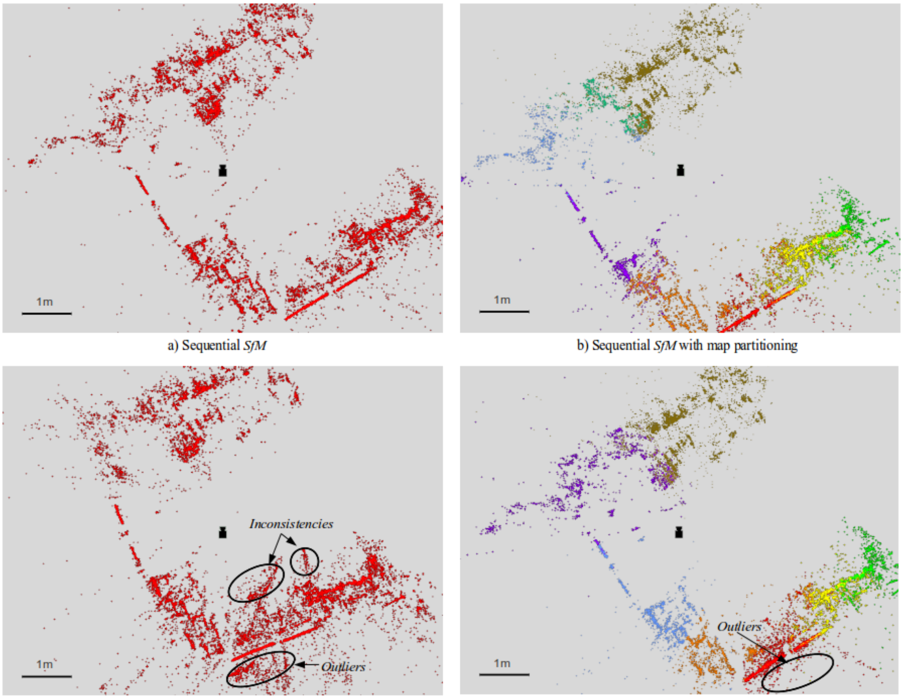
Figure 4.7: Bundle adjustment computation time (offline) with and without partitioning.

Additionally, we are interested in comparing the accuracy of the generated metric map. Due to the lack of a reliable metric to evaluate the map's quality, we have compared visually the different maps considering as ground truth the map obtained offline in the previous experiment (figure 4.8.a), which is the most accurate we can get. In the map obtained with PTAM (figure 4.8.c), we can appreciate some regions with depth errors and many outliers (e.g. landmarks detected behind physical walls). These inconsistencies appear as a consequence of the premature interruption of global BA that happens when a new keyframe is selected, which leads to data association errors and the subsequent loss of accuracy with the map size. On the contrary, the map obtained with our approach (figure 4.8.d) presents no inconsistencies and considerably fewer outliers than the unique map solution (figure 4.8.c). This results from the higher efficiency of the submap local optimization, which optimizes regions with highly correlated observations to produce locally accurate submaps.

The results shown in this section have been supported in several tests performed under different conditions: exploring different rooms, re-visiting previous maps, traversing a corridor, zooming to get more detail of the scene, etc. The reader may refer to `http://youtu.be/-zK05EcOjX4` for a video that illustrates the operation of our submapping approach with PTAM in different environments.

4.5 Discussion

This chapter presents an online metric-topological mapping technique which maintains a structure of local metric maps by grouping highly connected observations. Such local maps are obtained from graph cut, by grouping co-visible observations. This hybrid metric-topological structure improves the scalability of SLAM in two aspects: first, the system rules out unnecessary metric information to perform localization more efficiently; and second, it permits to approximate the global map op-



a) Sequential *SfM*

b) Sequential *SfM* with map partitioning

c) PTAM

d) PTAM with map partitioning

Figure 4.8: Top view of maps generated in our experiments. All the maps are composed of more than 400 keyframes and 22.000 landmarks. The different colors in b) and d) represent different submaps.

timization by a local optimization to reduce computational cost while maintaining the map consistency. Experimental results have demonstrated the potential of our approach to obtain efficient map representations in large environments, permitting a monocular SLAM system designed for small environments to operate in large scale. Furthermore, the topological arrangement of the map is useful for other tasks, as loop closure, global localization or navigation. A possible line of future work after this thesis may include exploiting the topological structure of the proposed mapping technique for loop closure and relocalization.

Chapter 5
A SLAM system for omnidirectional RGB-D sensors

Abstract

Simultaneous Localization and Mapping (SLAM) is a central problem for autonomous mobile robotics. It requires building a map from the sensor measurements at the same time that the robot is localized in such map. This chapter presents a new indoor SLAM solution employing an omnidirectional RGB-D device. The solution presented here is based on a hybrid metric-topological mapping approach consisting of a graph where the nodes are keyframes, corresponding to the omnidirectional (or spherical) RGB-D images, and the arcs represent the relative poses of pairs of keyframes. Each node is described through a plane-based map (PbMap), and localization is performed through PbMap registration. The map is optimized in a pose-graph framework applying dense pixelwise matching of the keyframes. This hybrid map is also structured in a second topological layer where closely related keyframes are clustered. Such higher level organization permits efficient re-localization and loop closure for optimizing the global consistency of the map.

5.1 Introduction

Omnidirectional images are traditionally defined as those images whose field of view (FOV) comprises 360° in the horizontal plane. They are also referred to as spherical images [Meilland *et al.*, 2010] since they can be warped on a sphere covering a large area (in this chapter we use these two terms with no distinction). Such images have some important advantages in computer vision and robotics, since problems as optical flow, or feature selection and matching are better conditioned. Furthermore, spherical vision provides a natural decoupling between rotation and translation, which is useful for localization in mobile robotics. These advantages have already been exploited during the last decades for scene modelling [Micušık *et al.*, 2003], vision-based navigation [Gaspar *et al.*, 2000], robot localization [Tamimi *et al.*, 2006; Menegatti *et al.*, 2006; Meilland *et al.*, 2011], visual odometry [Scaramuzza and Siegwart, 2008], place recognition [Ulrich and Nourbakhsh, 2000; Jogan and Leonardis, 2000], and SLAM [Kim and Chung, 2003; Rituerto *et al.*, 2010].

The availability of depth images is much more recent than for intensity ones (RGB) due to the latter development of dense depth perception. One solution for obtaining spherical depth images is omnidirectional LIDAR, as Velodyne [Glennie and Lichti, 2010]. However, the expensive price of this sensor (about $75000) prevents more extended applicability. A different strategy to obtain spherical RGB-D images is by using a rig of RGB cameras [Meilland *et al.*, 2010], where the depth is obtained from dense stereo matching [Hirschmuller, 2005]. In such case, this omnidirectional RGB-D device allows to build realistic representations of the world, permitting also accurate localization through dense image alignment [Meilland *et al.*, 2011]. However, the need to construct the RGB-D spheres offline puts an important limitation for its application in SLAM.

In this thesis we propose to use a rig of RGB-D sensors to obtain omnidirectional intensity and depth images at video frame rate (30 Hz). This approach presents some advantages with respect to the above ones like real-time acquisition, easy calibration and lower price (around $1800). On the other hand, the approach presented here is only valid for indoor environments. Outdoor environments could nonetheless be treated given that the depth can be obtained and that the scene contains planar surfaces. Regarding the first aspect, the main drawbacks of the RGB-D sensors used here is that they cannot compute the depth with direct sunlight and that they have a short useful range. These problems will likely be alleviated with the future versions of these sensors, but by now, they can only be avoided using more expensive sensors like Velodyne. Regarding the lack of planar structure, our approach can still work thanks to the pixelwise registration, however, the frame rate will drop in such case as this technique is considerably slower than PbMap registration. This sensor set-up is intended for quick scene reconstruction and for the creation of hybrid maps (metric-topological-semantic) for autonomous navigation. In this chapter it is used for SLAM employing the mapping approaches introduced in previous chapters of this thesis, to allow efficient operation through a compact and descriptive map. Concretely, the localization procedure is based on plane-based map (PbMap) registration, where a

PbMap descriptor is computed online for each omnidirectional image. The map consists of a set of keyframes which are selected when they provide new information about the scene, either when they observe an unexplored place or when the scene has changed considerably with respect to the previous observation. We perform some preliminary experiments in office and home environments confirming that despite the big volume of data acquired by the sensor, SLAM can still perform in real-time.

5.1.1 Related works

A variety of SLAM approaches have been presented for different sensors and conditions, a good introduction on these methods is given in [Durrant-Whyte and Bailey, 2006] and [Bailey and Durrant-Whyte, 2006]. In this chapter we focus on omnidirectional RGB-D SLAM, which has its own particularities with its pros and cons. Visual SLAM from omnidirectional cameras has already been investigated following the approaches based on the Extended Kalman Filter [Rituerto *et al.*, 2010], or pure topological SLAM [Goedemé *et al.*, 2007]. Previously, this source of data has been used for visual odometry with perspective omnidirectional vision [Tardif *et al.*, 2008] and with a catadioptric camera [Scaramuzza and Siegwart, 2008]. For the latter, loop closure was also proposed in [Scaramuzza *et al.*, 2010]. Topological mapping has been addressed with omnidirectional cameras like in [Menegatti *et al.*, 2002] where the authors employ a spatially semantic hierarchy. These images have also advantages for semantic inference and image classification [Oliva and Torralba, 2006], [Rituerto *et al.*, 2012].

A SLAM system only from omnidirectional depth images acquired with Velodyne was proposed by [Moosmann and Stiller, 2011]. Their *Velodyne SLAM* is based on Iterative Closest Point (ICP) registration [Chen and Medioni, 1992] which is performed on planar surfaces characterized by low uncertainty. This solution achieves nice point cloud representations with good global consistency over long trajectories in outdoor environments. Despite the registration stage is focused on planar surfaces, the compactness of such features is not exploited here where ICP is still performed using a classical pointwise cost function. As a result this is only useful with low frame rates or for offline mapping. Similarly to this work, we make use of planar patches for registration of RGB-D images, but we employ a compact description of them which abstracts from the 3D points (pixels). This strategy allows to perform faster inter-frame registration in real-time (30 Hz). Also, our technique does not require any initial estimation and furthermore, it can register frames further apart, so that it can be applied for re-localization and loop closure detection.

In the context of omnidirectional RGB-D data, probably the first mapping system to create large models is that of [Meilland *et al.*, 2010], which is intended for urban autonomous navigation. This work was extended in [Meilland *et al.*, 2011] to create dense representations of large environments which are used later for robot localization with a regular monocular camera. Recently, another omnidirectional RGB-D sensor rig, very similar to the one presented in this thesis, was built by [Schwarz and Behnke, 2014] to perform navigation in rough terrain. Besides mapping and navigation, omnidirectional RGB-D images provide very rich information for SLAM. However, it still constitutes an open problem where the main question is how to manage the big volume of data captured by the sensors.

The approach to SLAM presented here is based on PbMaps, which are used as a descriptor to solve quickly the registration of omnidirectional RGB-D frames. Other SLAM approaches can be found in the literature based on planar patch features extracted from a rotating laser scanner [Weingarten and Siegwart, 2006; Pathak *et al.*, 2010a]. These employ a probabilistic framework to build a map of planar patches which is updated at a low frame rate limited by the frequency of the sensor. The solution proposed here differs from those in a few aspects, mainly in the mapping approach, which is based on a nested structure of keyframes with different topological levels to allow for large scale operation with efficient re-localization and loop-closure. Also, our localization strategy takes into account the spatial relations of neighbouring planes for higher robustness, and finally, the depth and intensity information is exploited through pixelwise registration in a back-end process to refine the keyframe's poses through pose-graph optimization.

5.1.2 Contribution

We present a new sensor rig to capture omnidirectional RGB-D images at video rate (30 Hz), and a new SLAM system employing such omnidirectional RGB-D data. This novel device has important prospective applications for scene reconstruction and mobile robotics, including SLAM. The SLAM approach proposed here is based on hybrid metric-topological mapping, where localization is achieved by PbMap registration. Its main advantage comes from the rapid online registration of spherical RGB-D images using a compact plane-based description of the scene. The map is organized in a metric-topological structure of keyframes which is rearranged dynamically as new observations are available. This map structure permits to perform efficient re-localization and loop closure with sub-linear computation time on the map size. Also, the global consistency of the map is improved through pose-graph optimization, for which the connections between keyframes are refined by dense RGB-D alignment. This dense alignment method is a modified version of a previous method to take into account occlusions and thus, be able to register frames that are further apart.

Next, we present the details of our sensor set-up and the acquisition of spherical images, analysing the pros and cons between the different alternatives to obtain spherical RGB-D images. Then, we describe our SLAM approach (section 5.3), where we detail: the localization technique based on fast registration of PbMaps and dense RGB-D alignment; the mapping process, which is based on a hierarchical structure of keyframes; and the loop closure approach. Some preliminary experiments are presented next within home and office environments (section 5.4). Finally, we expose the conclusions of this work and advance some lines of future research.

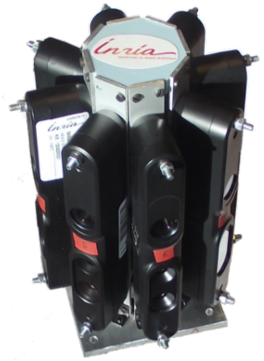


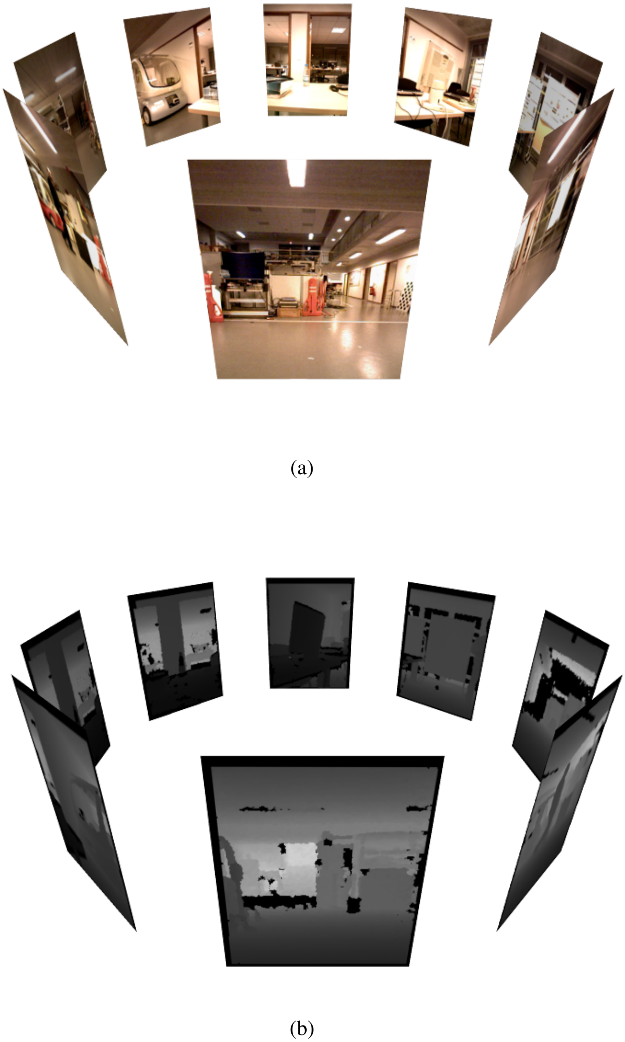
Figure 5.1: Omnidirectional RGB-D camera rig.

5.2 Omnidirectional RGB-D device

5.2.1 Sensor set-up

Our device for omnidirectional RGB-D acquisition is composed of 8 Asus Xtion Pro Live (Asus XPL) sensors, which are mounted vertically in a radial configuration at an angle of 45° as shown in figure 5.1. An example of the images captured by this sensor is shown in figure 5.2. This device is connected to a computer through two PCIe cards with 4 USB ports each. This set-up permits to capture 360° field of view in the horizontal plane with no overlap among sensors, avoiding problems of interference in the infrared images (the vertical FOV of the Asus XPL sensor is 45°). The vertical FOV of our device correspond to the horizontal FOV of Asus XPL, being 63° and its maximum resolution is 3840 x 640 (2.46 Mpx). The whole system works at 30 Hz without synchronization between the different sensors. The latter is not an issue here since the system is mounted on a robot moving at a maximum speed of 1 m/s, what permits to approximate the reconstructed spherical images considering that the 8 pairs of RGB and depth images are taken simultaneously.

Previous alternatives to capture omnidirectional depth or RGB-D images include 3D LIDAR (like e.g. Velodyne), and multicamera rigs. Table 5.1 shows a comparison between these options and the device proposed here. Regarding the 3D LIDAR, it still constitutes a very expensive option, so that its use has been mainly limited to complex projects in the field of autonomous cars. This sensor can provide images of about 1 Mpx at 15 Hz with a vertical FOV (26.8°). This reduced vertical FOV is more amenable for outdoor applications. Also, in order to obtain RGB-D images, the radiometric information must be captured with a separate sensor, requiring calibration between both [Mirzaei *et al.*, 2012]. A different option to obtain spherical RGB-D images through a rig of RGB cameras was presented in [Meilland *et al.*, 2010] which reconstructs the scene based on dense stereo matching. In this way, the corresponding intensity and depth images are available with no need of further geo-



(a)

(b)

Figure 5.2: Images captured by the omnidirectional RGB-D sensor: a) RGB and b) depth.

metric corrections. However, this technique requires well textured scenes to produce consistent depth images, which besides need to be computed offline. Here, the main advantage of our RGB-D sensor is the combination of a more affordable price, with big field of view and a high frequency of acquisition.

Table 5.1: Characteristics of different spherical depth image devices.

Acquisition	*LIDAR*	*Stereo-rig*	*RGB-D-rig*
Information	Depth	RGB-D	RGB-D
Price	75000$	7500$	1800$
Frequency	5-15 Hz	30-60 Hz	30-60 Hz
Field of view	26.8°	125°	63°
Range	70 m	60 m	5 m
Accuracy	2 cm	2 cm / m	1 cm / m
Outdoor	Yes	Yes	No
Indoor	-	Yes	Yes

5.2.2 Calibration

The proposed rig of RGB-D sensors needs to be calibrated in order to put the data in the same reference frame. For that, both intrinsic and extrinsic parameters must be estimated. First, the intrinsic parameters of the 8 RGB cameras are estimated independently using a chequerboard pattern, following section 2.2.1. Then, an intrinsic model is estimated for each depth camera to reduce the bias of depth measurements [Teichman *et al.*, 2013] (see section 2.2.2). A combined methodology to calibrate each sensor, including intrinsic parameters of RGB and depth cameras together with the extrinsic parameters between them[1] was not applied here because this type of solution does not take into account the bias of the depth measurements. Besides, we found that the extrinsic parameters provided by the sensor manufacturer are good enough for our application. Once the intrinsic calibration is done, the extrinsic calibration between the 8 RGB-D sensors is obtained applying the method proposed in section 2.3.2.

5.2.3 Spherical representation

The chosen spherical representation has the advantage of modelling the image isometrically, i.e. the same solid angle is assigned to each pixel. This permits to apply directly some operations, like point cloud reconstruction, photoconsistency alignment or image subsampling. To build the images, the sphere $\mathbb{S}^2$ is sampled according to the resolution of our device, so that the equator (θ direction) contains 3840 divisions in the range $[0, \pi]$, and the (ϕ direction) is sampled keeping the same angular proportion, so that it contains 1920 divisions in the range $[-\pi/2, \pi/2]$. Actually, since the sensor does not observe the full range in ϕ, we only store the useful range which corresponds to a vertical FOV of in the range $[-\pi/6, \pi/6]$.

For spherical warping, a virtual sphere with the above sampling and radius $\rho = 1$ (unit sphere) is used to project the sample points into image coordinates (u, v), (see

[1] www.mrpt.org/tutorials/programming/miscellaneous/kinect-calibration/

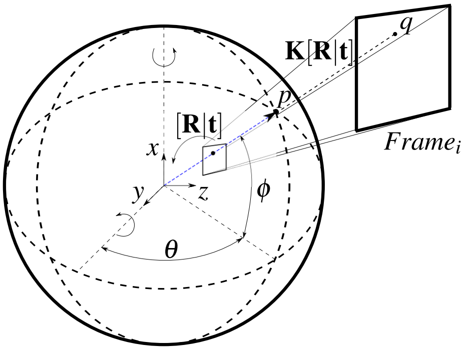


Figure 5.3: Spherical image construction

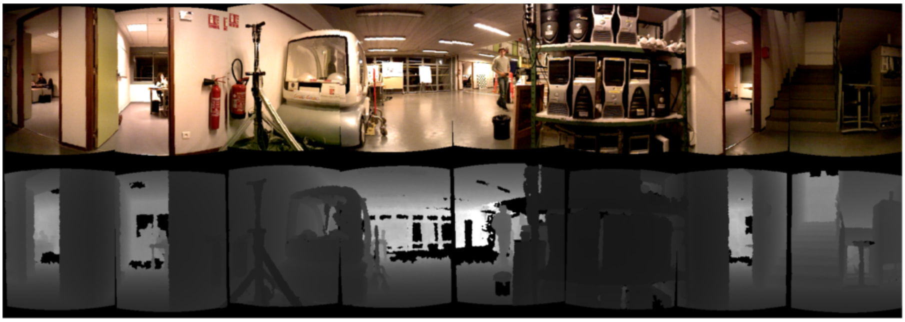


Figure 5.4: Omnidirectional RGB and depth images acquired by our RGB-D camera rig.

figure 5.3). For that, the extrinsic calibration of each sensor is taken into account. Thus, a point $\mathbf{p}$ in $\mathbb{S}^2$ is parametrized in $\mathbb{R}^3$ as $\mathbf{p} = [x, y, z]^T$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \rho\ sin(\phi) \\ \rho\ cos(\phi)\ sin(\theta) \\ \rho\ cos(\phi)\ cos(\theta) \end{bmatrix} \tag{5.1}$$

The point $\mathbf{q} = (u, v)$ on image coordinates is found by applying perspective projection to $\mathbf{p}$, through the homogeneous matrix $\mathbf{M} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$, where $\mathbf{K} \in \mathbb{R}^{3x3}$ is the camera projection model and, $[\mathbf{R}|\mathbf{t}] \in \mathbb{SE}(3)$ is the relative position of the camera with respect to the sphere reference frame (extrinsic calibration). Nearest neighbor interpolation is used to assign the intensity and depth values to the respective spherical coordinates. Figure 5.4 shows an example of the RGB and depth spherical images obtained from this technique.

In order to obtain the point cloud from the spherical representation the equations 5.1 are applied, substituting ρ by the measured depth and the values of θ and ϕ by their corresponding image location.

5.3 SLAM approach

The SLAM approach presented here is based on a map of keyframes corresponding to selected omnidirectional RGB-D images which are organized in a topological structure of local submaps. The keyframes are described through a compact PbMap, which is used for efficient localization and loop closure. Our SLAM solution can be described through two main concurrent processes: a front-end localization process which tracks the sensor pose from the streaming images, and a back-end process for map construction and loop closure optimization. These two processes are illustrated in figure 5.5, and are described separately below.

5.3.1 Localization

The tracking problem is addressed here by registering PbMaps that are extracted from each omnidirectional RGB-D image. Figure 5.6(a) shows the point cloud built from an omnidirectional RGB-D observation, with the coloured patches of its PbMap descriptor superimposed in figure 5.6(b). The PbMap structure and its registration procedure were described in chapter 3. Differently from other works that register planar patches like [Weingarten and Siegwart, 2006; Pathak *et al.*, 2010a], our approach takes into account the geometric relationships between the different planes in the scene, making this compact descriptor much more reliable, and thus, making it suitable for loop closure and re-localization. This registration technique also provides the

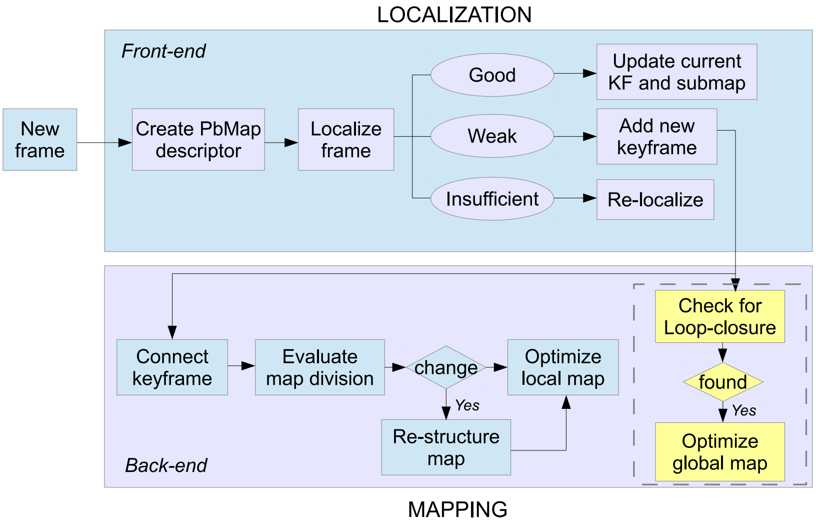
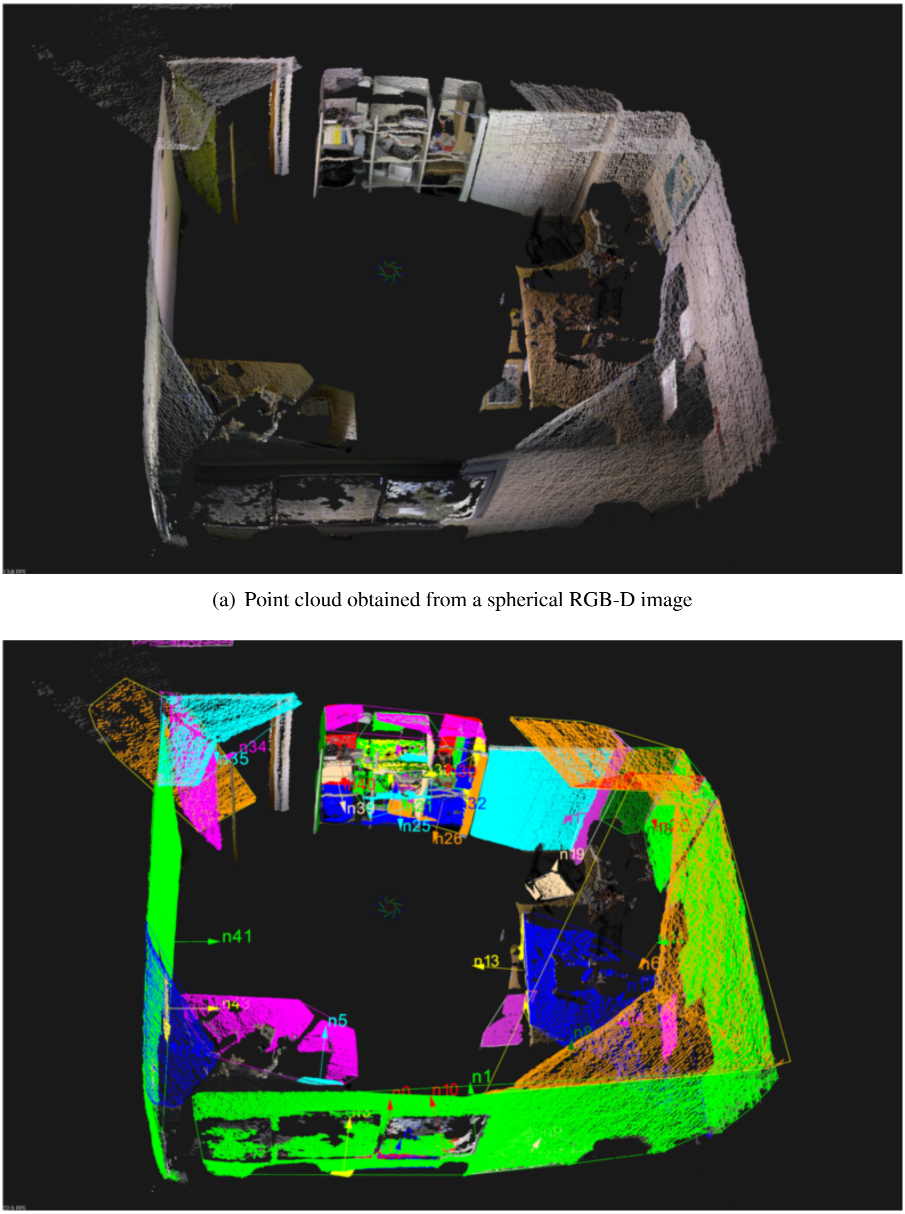


Figure 5.5: Block scheme of our keyframe based metric-topological SLAM using an omnidirectional RGB-D sensor.



(a) Point cloud obtained from a spherical RGB-D image

(b) Point cloud with segmented planes superimposed.

Figure 5.6: Point cloud visualization of the spherical image from fig. 4.

covariance (uncertainty) of the relative pose between the frames registered, which is useful to optimize the map through pose-graph optimization.

The whole process of localization is illustrated in the upper block of figure 5.5. It starts when a new frame is acquired, by computing the PbMap descriptor for that frame. Then, it is registered to the closest keyframe, for which a reference to that keyframe is always kept in this process (during exploration this generally corresponds to the last keyframe selected). This registration can result in three different outcomes depending on some heuristic parameters: a) "good registration", when more than 70% of the scene planes are matched; b) "weak registration" when the number of matched planes is between 70% and 30%; and c) "insufficient registration", when less than 30% of the planes are matched. These thresholds were chosen heuristically after some tests trying different values. Then, if the registration with the closest keyframe is not good, this process is repeated sequentially until a "good registration" is achieved with a keyframe of the current local map, in which case, the index of the closest keyframe is updated. If the best localization is weak, then we can assume that the sensor explores a new part of the scene and the current frame is selected as a new keyframe. Finally, if registration is insufficient, a re-localization algorithm is launched that looks for the current image in the whole map, starting from the nearest submaps to the last tracked position.

The localization may also be refined by applying a dense alignment method based on the consistency of both pairs of intensity and depth images, in a similar way as it is described in [Gokhool *et al.*, 2014]. This technique is more accurate than PbMap registration, since it makes use of all the information in the RGB-D images, but on the other hand it is considerably slower (about 2 orders of magnitude). Therefore, this refinement is applied only to the keyframes to improve the registration with their neighbour ones. This is carried out in the background inside the mapping process, concretely in the block *Connect keyframe*, but it is described here since it is intrinsically related to localization.

This dense registration technique minimizes the error from two different metrics that measure the differences between the reference image and the target one, where the latter is warped according to the relative pose $T(\mathbf{x})$ which is estimated iteratively. The photoconsistency error function is given by

$$\mathscr{F}_I = \sum_{i=1}^{n} \eta_{HUB}\Big(I(w(T(\mathbf{x});P_i^*)) - I^*(w(T(0);P_i^*))\Big)^2 \tag{5.2}$$

where I and I^* are the target and reference images respectively, $w(\cdot)$ is the warping function that projects a 3D point P_i^* from the reference image onto the target sphere according to the relative pose $T(\mathbf{x})$ between them, where $\mathbf{x}$ is a minimal parametrization of the pose in the Lie algebra $\mathfrak{se}(3)$ and the operator $T(\cdot)$ represents the exponential map to $\mathbb{SE}(3)$ (see appendix B). Finally, η_{HUB} is a robust weighting function given by Huber's M-estimator [Huber, 1981]. This robust estimator contributes to reduce the effect of outliers with large intensity differences, which may arise as a result of specular reflections for instance. On the other hand, the depth consistency minimizes the cost

$$\mathscr{F}_D = \sum_{i=1}^{n} \eta_{HUB} \left(D(w(T(\mathbf{x}); P_i^*)) - \|T(\mathbf{x})P_i^*\|\right)^2 \tag{5.3}$$

where D is the depth source image and $\|\cdot\|$ is the L2-norm operator. This cost function is equivalent to the formulation of point-to-plane ICP with projective lookup.

The optimization of such RGB-D image alignment is computationally demanding because all the pixels have to be reprojected along several iterations of the method. In order to speed-up the estimation, we only consider the salient pixels of both intensity and depth images (i.e. pixels with high gradient on the image) since the rest of the pixels provide little or no information. The resulting speed-up is directly related to the proportion of pixels used, which in our case has been manually set to 10 %, qualitatively producing similar registration results in our experiments.

The cost functions above have been used for registration of both: projective RGB-D images like those captured by Kinect [Kerl *et al.*, 2013b], and spherical RGB-D images as in [Meilland *et al.*, 2010]. The only difference between them lies in the warping function which is specific for each case. In any case, the images to be registered are supposed to be taken from very close positions, so that the possible occlusions are neglected. This is not the case here, where keyframes which are further apart are to be registered, so that the co-visibility of these should be enough to be able to compute the alignment, but occlusions must be handled as they may introduce important deviations in the registration. In order to cope with this situation, a depth-buffer of the projected pixels (as in [Lieberknecht *et al.*, 2011]) is employed here to discard the occluded points from the sums above.

Both cost functions for intensity and depth consistency depend on the relative pose $\mathbf{x}$ between the frames, but they have different scales. Several methods can be found in the literature to weight these two functions. Here, we follow the work of [Kerl *et al.*, 2013a] to weight the intensity and depth errors within a probabilistic approach depending on the error variances, which are assumed to be independent. However, since both errors are considered to be independent, we employ different Huber estimators for each, instead of using a common bivariate weight as in [Kerl *et al.*, 2013a]. Thus, the resulting least squares problem corresponds to a robust maximum likelihood estimation (see appendix A).

5.3.2 Mapping

The map creation process consists of building a network of keyframes described by individual PbMaps, where each keyframe is connected to those keyframes near-by with which relative localization is possible through PbMap registration. The map is organized in a structure of local submaps which contain highly related keyframes. These submaps are also connected themselves when there exist connections among their keyframes. Figure 5.7 shows a representation of this structure, where we can see the two levels of topological information (local and global). This mapping strategy is based on previous works focused on scalable mapping and navigation in complex environments [Blanco, 2009].

The lower block of figure 5.5 depicts the different actions carried out by our mapping approach. Thus, when a new keyframe is provided by the localization process, the current local map is updated to include this keyframe, establishing also the new connections with the registered keyframes in the current submap and its one-connected submaps, for what dense RGB-D registration is also applied. Each keyframe connection stores a relative pose in $\mathbb{SE}(3)$ and its 6×6 covariance matrix which are obtained from the registration stage, storing also a scalar which represents the co-visibility of the pair of keyframes. Following [Blanco *et al.*, 2006] we call this value sensed-space-overlap (*SSO*), which we define as

$$SSO = \frac{A_{shared}}{A_{shared} + A_{diff}} \tag{5.4}$$

where A_{shared} represents the total area of the matched planar patches, and A_{diff} represents the sum of the areas of the non-matched patches (for both measures we take the average of the two PbMaps,). Thus, the $SSO \in [0,1]$, where $SSO = 1$ when all the planes in both PbMaps are matched, while $SSO = 0$ when the frames are not registered (no keyframe connection). The *SSO* is used here to organize the map into a higher level topological structure following the methodology presented in chapter 4. Thus, after a new keyframe is added to the map, the submap arrangement is re-evaluated to maintain a structure of local maps containing highly related keyframes (this may result in a larger, equal, or even smaller number of local maps depending on the new keyframe connections). This process affects the current local map and its first order neighbors, and it is performed very quickly since the map division strategy is very fast (see chapter 4) and re-organizing the map only implies re-arranging the keyframe indices.

The concept of sensed space overlap also permits to define what we call the most representative keyframe of a local map. This keyframe corresponds to the one with the highest index of shared information *ISI*, which is computed for each keyframe as the sum of the *SSO* with all its connections. The *ISI* can be intuitively seen as the connection score of a keyframe within a submap. For example, considering a local map corresponding to a single room of a building, the highest *ISI* will generally correspond to a keyframe in the center of the room which observes most of the planes and with the minimum occlusions. Identifying the most representative keyframe has two different advantages: it permits to summarize the information of the map with a reduced number of keyframes, and as a consequence, problems like re-localization or loop closure can be performed more efficiently by considering the most representative keyframes first.

Each submap stores also its pose with respect to a global coordinate system, and each keyframe stores its pose relative to the submap's reference system. Such poses allow to represent all the keyframes unequivocally in a common reference frame. This

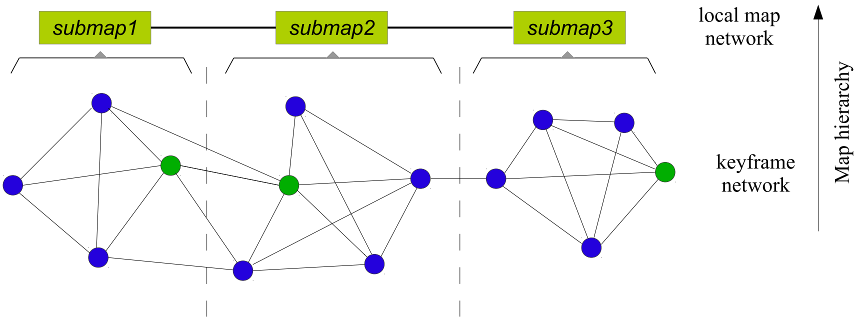


Figure 5.7: Hybrid map structure with two topological layers: a higher layer where each node represents a local map of highly related keyframes, and a lower layer with a network of keyframes. The most representative keyframe of each local map (the one with highest *ISI*) is coloured in green.

is useful to build consistent metric maps, e.g. a single point cloud or PbMap combining the information of all keyframes into a global map (see figure 5.9). For that, the poses of both the submaps and the keyframes are obtained from pose-graph optimization of the global and local maps respectively, taking into account all its keyframe connections and their covariances. This graph optimization is carried out using the publicly available library *g2o*[2] [Kummerle *et al.*, 2011]. For that, every time a new keyframe is added to the map, whether the topological structure is modified or not, the local map is optimized to update the keyframe positions. Also, loop closure is searched for with every new keyframe, and if it is found, the map division is rearranged and the relative poses of the local maps are updated also through pose-graph optimization similarly as it is done for the keyframe poses inside a local map. Such loop closure algorithm is detailed separately in the next section.

5.3.3 Loop closure

Some loop closure approaches have been proposed in the literature which are specially suited for omnidirectional images, like [Chapoulie *et al.*, 2011] which is based on the well known bags of visual words, or [Oliva and Torralba, 2006] which is based on the registration of a global image descriptor. By employing PbMap registration also for loop closure, we reduce the computation burden with respect to the alternatives above, while we maintain the coherence in our SLAM approach which relies mainly on a geometric description and thus it can be applied also to range images. Note however that our loop closure strategy can be combined with those above to gain in robustness.

[2] `www.openslam.org/g2o.html`

The loop closure search is carried out with every new keyframe by identifying the most representative keyframes of the local maps which are nearer to it (excluding the current local map and its first order neighbors which are constantly checked for SLAM). In order to estimate the most likely locations for a loop closure, we compute the relative pose between the current keyframe and each local map together with its covariance (this is done through pose composition among the different reference frames). Then, the ratio between the squared root of the maximum eigenvalue of the covariance of the relative translation and the norm of such translation (i.e. the Euclidean distance), provides a comparative measure of how likely is the current frame to be near the local map being evaluated. Arranging such measures in decreasing order provides the order in which the different local maps are checked for loop closure. This strategy results in sublinear loop closure computation with respect to the map size.

Once the search order has been established, loop closure is tackled in a similar manner to the re-localization problem by registering the PbMap descriptor of the current frame with others from different local maps. If a match is found (the loop is closed), new keyframe connections are searched between the current local map and the one with which the loop has been closed. Then, the pose-graph containing the poses of the different local maps is optimized to include the new constraints of the loop closure in the global map. This optimization is carried out in a similar way as for the keyframes of a local map using *g2o* [Kummerle *et al.*, 2011].

5.4 Experimental validation

This section presents some preliminary experiments to validate our SLAM system. These experiments are carried out with a wheeled robot with planar movement (see figure 5.8), though the SLAM approach is designed to work with 6 degrees of freedom. The robot has an on-board computer which performs all the computation with an Intel i7 processor with 8 cores at 3.1 GHz and 8Gb of memory. In our experiments we employ a reduced resolution of the omnidirectional RGB-D images with 960x160 pixels, since higher resolutions do not affect significantly the plane segmentation results and they have a higher computational cost. The depth images captured by the sensor are corrected as explained in the section 2.2.2, such correction takes around 2 ms per omnidirectional image. Several sequences are taken exploring different home and office environments, where the robot is remotely guided by a human at a maximum speed of 1 m/s.

5.4.1 Fast scene registration

The main feature of our SLAM system is the fast registration of omnidirectional RGB-D images, which is used for camera tracking, re-localization, and keyframe selection. In this section we present experimental results comparing the performance of PbMab based registration with other registration approaches like ICP and dense intensity and depth alignment. Our registration approach requires building the PbMap

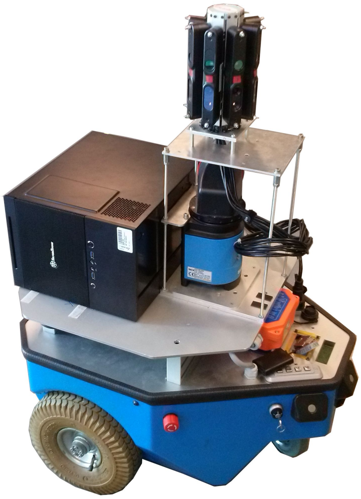


Figure 5.8: Robot with the omnidirectional RGB-D sensor.

descriptors from the spherical RGB-D images, which implies the segmentation of planar surfaces from the images. Such segmentation is performed efficiently through region growing (see chapter 3), being the most demanding task for registration. This stage is also parallelized to exploit our multi-core processor to segment the planes of the spherical image in less than 20 ms. PbMap matching requires much less computation, in the order of microseconds.

Furthermore, both ICP and dense alignment also need a previous preparation to compute the spherical point cloud and the spherical images, respectively, before computing the matching. Table 5.2 presents the average computation time of these three methods for spherical RGB-D image registration, calculated from 1000 consecutive registrations (odometry). For that, both ICP and dense alignment are performed using a pyramid of scales for robustness and efficiency. In this table, we can see how the registration based on PbMap is two orders of magnitude faster than the other two alternatives.

Table 5.2: Average RGB-D sphere registration performance of different methods (in seconds).

	PbMap	*ICP*	*Dense*
PbMap construction (s)	0.019	-	-
Sphere construction (s)	-	0.010	0.093
Matching (s)	10^{-6}	1.53	2.12
Total Registration (s)	0.019	1.54	2.22

Besides the low computational burden, another important advantage of our registration technique with respect to classic approaches like ICP or dense alignment is that we do not require any initial estimation. Thus, we can register images taken further away, while ICP and dense alignment are limited to shorter distances without a good initial estimation (i.e. considering the identity as the initial transformation). This fact is also illustrated in table 5.3, which shows the average maximum Euclidean distance between the registered frames of the previous sequence. For that, each frame is registered with all the preceding frames until tracking is lost, selecting the last registered frame as the furthest one. Also, our method is better suited to dynamic environments where humans or other elements are constantly moving, since the large planar surfaces taken into account for registration are generally static. Home and office environments are a good example for that, where the humans change their pose, and also the poses of some objects like chairs, but where the scene structure remains unchanged.

Table 5.3: Average of the maximum distance for registration with different methods.

	PbMap	*ICP*	*Dense*
Registration dist. (m)	3.4	0.39	0.43

The registration of RGB-D images through PbMap permits to perform odometry estimation of the robot trajectory efficiently. This is done simply by registering the current frame to the previous one (see the video at `www.youtube.com/watch?v=8hzj6qhqpaA`). Figure 5.9 shows the trajectory followed by our sensor in one of our exploration sequences in a home environment together with the point clouds from each spherical image superimposed. The consistency of the resulting map indicates that each sphere is registered correctly with respect to the previous one, though yet, we can appreciate the drift in the trajectory which comes as a consequence of the open loop approach. This qualitative experiment shows that despite the compact information extracted for fast registration of the spherical images, the accuracy of registration is still good for many applications.

5.4.2 Keyframe-based SLAM

The above results for fast registration are exploited here to perform SLAM based on a multilayer metric-topological map of keyframes. The map is built concurrently while the robot explores different environments, including home and office environments. These experiments are basically a proof of concept for a new robust and efficient SLAM solution from omnidirectional RGB-D data. To our knowledge, this is the first SLAM system using such kind of data and thus, a comparative study cannot be provided here.

The operation of our SLAM approach is shown with a video in `www.sites.google.com/site/efernandezmoral/projects/rgbd360`, where we can see how the map is built while the robot explores the scene, by adding new keyframes when

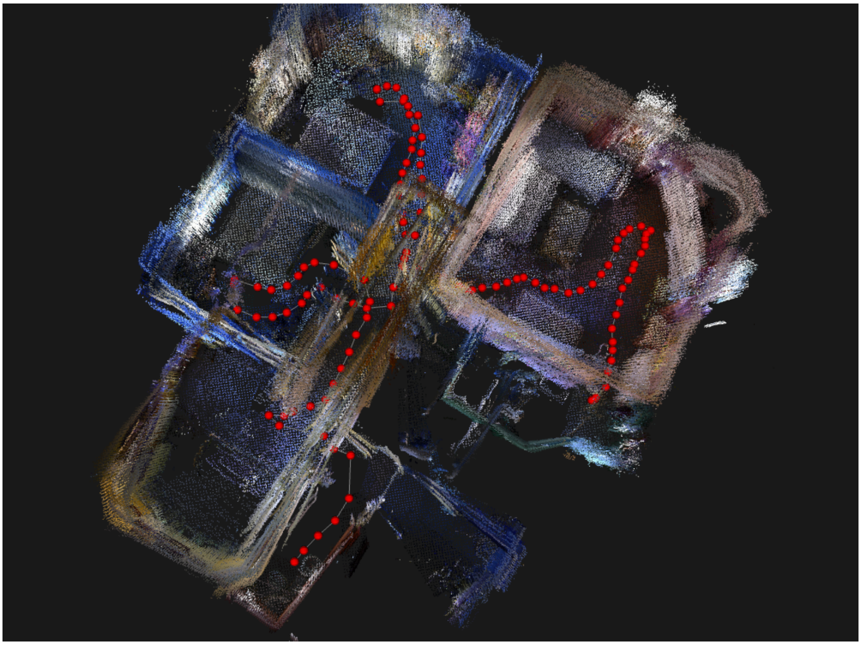


Figure 5.9: Trajectory of the sensor in a home environment composed of different rooms (the path is about 36 m).

they provide new information of the scene. A snapshot from this video is shown in figure 5.10 showing the map as a set of superimposed point clouds extracted from keyframe locations, which are shown with sphere objects. Such keyframes are connected to other nearby keyframes with which they were registered, forming a network which is optimized as a pose-graph. The spheres are shown with different colours representing the different local maps. As we can see, the different local maps correspond to meaningful areas of the environment (i.e. different rooms), making this representation suitable for topological navigation.

This representation is also well adapted to large scale SLAM operation since only a local portion of the whole map is managed as the robot moves around the scene. However, experiments on large scale are not shown here due to limitations in the environment where we had access during this thesis (i.e. small buildings). Such a work is left for some future research. From this proof of concept experiments we also see that the whole map is highly consistent thanks to the loop-closure mechanism.

Another advantage of our approach that we corroborate in our experiments is the suitability of the maps for variable illumination. This is a direct consequence of using mainly geometric information extracted from depth images which do not depend on the available light (with the exception of direct sunlight). If the proposed representation is to be used for more complex tasks which may require the intensity information (e.g. object recognition), the map can be easily adapted to take new keyframes when the lighting is considerably different to the previous time when that area was mapped, like during day and night.

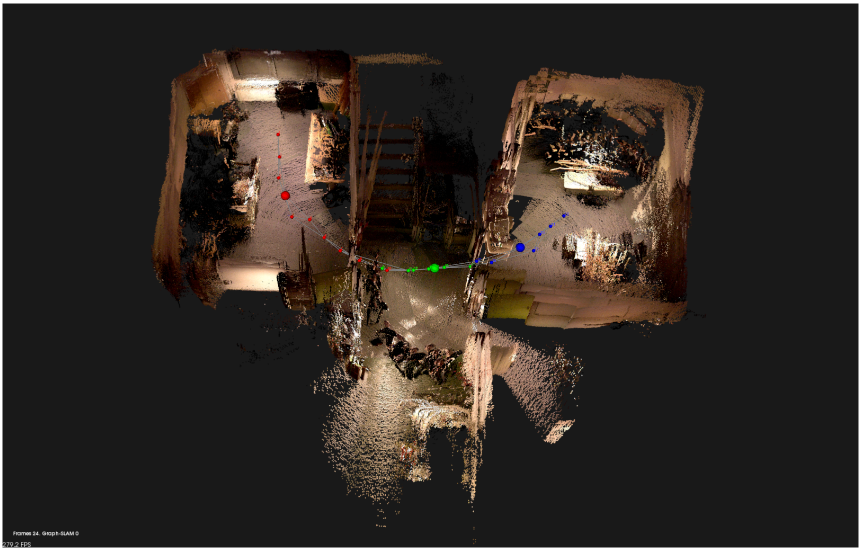


Figure 5.10: Keyframe-map of an office environment. The spheres represent the location where the keyframes were taken. The large spheres are the most representative keyframes of each local map, where different colours are used to represent such local maps.

5.5 Discussion

A novel sensor set-up has been proposed here for online acquisition of spherical RGB-D images. This approach has advantages over other alternatives used today in terms of accuracy and real-time spherical image construction for indoor environments, which are specially interesting for mobile robotics. A calibration method for such device is presented, which takes into account the bias of each sensor independently. The proposed calibration method does not require any specific calibration pattern, taking into account the planar structure from the scene to cope with the fact that there is no overlapping between sensors. In order to demonstrate the potential of this device, we show how these images can be registered in real-time by extracting and matching planar surfaces.

The proposed map structure has several advantages with respect to previous approaches in the literature: first, the map stores complete information about the scene in a compact fashion; second, it permits fast keyframe registration through the compact PbMap descriptors for localization and loop closure, while dense registration is applied to refine the relative poses between keyframes; third, the map is maintained as a pose-graph which is optimized locally when new keyframes are added, and globally when loop closure is detected; and finally, the topological structure can also be used to define attributes of the scene like rooms, or to recognize places.

The next step in our future research is focused to dynamic SLAM in scenarios that change constantly (e.g. presence of people moving, who also modify the objects present in the environment). For that, we plan to extract semantic cues in the scene that will be used for detecting changes in the scene, and so to update the existing map when necessary.

Chapter 6
Conclusions

This thesis has addressed different problems related to the topic of localization and mapping for mobile robotics. The research community has dedicated important effort to this topic and an extensive literature can be found around it. However, most approaches have still important limitations, mainly to cope with large scale and dynamic environments, and to work in a wider range of conditions and scenarios. In this context, several contributions have been presented in this thesis for calibrating sensor rigs, for efficient and compact map representations, and for fast and robust localization in such maps.

Localization and mapping in mobile robotics is often addressed using a combination of sensors, in which case, these must be calibrated to refer all the data to a common frame of reference. The particular problem of calibrating a rig of range sensors has been previously solved only for very particular conditions. In this thesis, we have proposed a new methodology that permits to calibrate any combination of 2D and 3D range sensors in arbitrary configurations from the observation of common planar surfaces. This methodology is easy to apply, not requiring any special calibration pattern, and it is applicable to different configurations of mobile robots and autonomous cars.

We have also presented a new mapping approach based on planar surfaces which can be easily segmented from range or RGB-D images. This plane-based map (PbMap) is particularly well suited for indoor scenarios, and has the advantage of being a very compact and still a descriptive representation which is useful to perform real-time place recognition and loop closure. A fast localization approach has been proposed to register contexts of planes by matching planar features taking into account their geometric relationships. This solution performs significantly faster than previous approaches. Also, a hybrid mapping strategy has been presented to deal with large scale SLAM and with navigation in complex environments. This approach organizes the map into local maps with highly related observations, permitting the abstraction of metric information unnecessary at the current robot location. For that, the map is dynamically organized in a metric-topological structure according to the sensor observations. Efficient large scale SLAM operation has been demonstrated in monocular SLAM.

Finally, a SLAM approach is presented for omnidirectional RGB-D data, integrating several advances achieved along this thesis. A new device was conceived for gathering this type of images at high frame rates (30 Hz) which combines several structured-light sensors. This device has important advantages for navigation and SLAM with respect to previous alternatives as: lower cost, large field of view, and high observation frequency. This SLAM system is based on summarizing the rich information gathered by the sensor in a compact sketch of planar surfaces (PbMap), which is structured in the metric-topological mapping based on keyframes to permit real time SLAM operation. Future work is envisaged for this new sensor and the SLAM approach to adapt better to dynamic environments by integrating semantic information about the scene.

Appendices

Appendix A
Maximum Likelihood Estimation and Least Squares

This appendix describes the theory of Maximum Likelihood Estimation (MLE) applied to problems where the observable data are modelled by a likelihood function following a Gaussian distribution, and presents its solution by common least squares optimization.

Maximum Likelihood Estimation

Maximum Likelihood Estimation refers to a method of estimating the parameters θ of a statistical model given by a likelihood function on some observable outcome $\mathbf{x}$. Such likelihood function expresses the probability of a measured sample for some given parameter values $\mathscr{L}(\theta|\mathbf{x}) = P(\mathbf{x}|\theta)$. When this probability follows a normal distribution, which is a very common assumption in mobile robotics and computer vision and is also employed along this thesis, the estimation problem coincides with the solution of weighted least squares minimization. To arrive to this result, let's consider the following Gaussian distribution

$$P(\mathbf{x}|\theta) = \frac{1}{\sqrt{(2\pi)^k|\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right) \tag{A.1}$$

defined by the mean μ (the unobservable true value of measured observation which depends on the model parameters θ) and the covariance Σ, where k is the dimension of the observation $\mathbf{x}$. In the general case where a series of measurements from $\mathbf{x}_1,\dots,\mathbf{x}_n$ are available, the likelihood of the parameters is expressed as

$$P(\mathbf{x}_1,\dots,\mathbf{x}_n \mid \theta) = \prod_{i=1}^{n} P(\mathbf{x}_i|\theta) \tag{A.2}$$

and thus, the maximum likelihood for the parameters $\hat{\theta}$ comes from

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \prod_{i=1}^{n} P(\mathbf{x}_i|\theta) \tag{A.3}$$

The result of this maximization is the same when applied to the log-likelihood

$$\begin{aligned}
\hat{\theta} &= \operatorname*{argmax}_{\theta} \left(\log \prod_{i=1}^{n} P(\mathbf{x}_i|\theta) \right) && \text{(A.4)}\\
&= \operatorname*{argmax}_{\theta} \sum_{i=1}^{n} \log P(\mathbf{x}_i|\theta) && \text{(A.5)}\\
&= \operatorname*{argmin}_{\theta} \sum_{i=1}^{n} (\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu) && \text{(A.6)}\\
&= \operatorname*{argmin}_{\theta} \sum_{i=1}^{n} \mathbf{r}^T \Lambda \mathbf{r} && \text{(A.7)}
\end{aligned}$$

which is equivalent to a weighted least squares problem, where the residual $\mathbf{r} = \mathbf{x} - \mu$ and the weights are given by the information matrix $\Lambda = \Sigma^{-1}$.

Least squares

This section details the optimization of the least squares form of the cost function $F(\mathbf{m})$, defined as the following quadratic form:

$$F(\mathbf{m}) = \mathbf{r}^\top \Lambda \mathbf{r} \tag{A.8}$$

where $\mathbf{r} = \mathbf{r}(\mathbf{m})$ is the vector of errors or *residuals*, a measure of the mismatch between the prediction and the observation, Λ stands for the information matrix, i.e. the inverse of the covariance matrix, and $\mathbf{m}$ are the unknown parameters to estimate. The information Λ is usually assumed to be independent on the parameters and the error is a known function of them $\mathbf{r}(\mathbf{m})$.

The minimum of the above cost function is found by restricting that its Jacobian must be zero

$$\left.\frac{\partial F(\mathbf{m})}{\partial \mathbf{m}}\right|_{\hat{\mathbf{m}}} = 0 \tag{A.9}$$

This problem has a closed form solution when the residuals are linear functions of $\mathbf{m}$. In the general case where the residuals do not depend linearly on the parameters this process can be solved iteratively, providing for each iteration a small increment $\Delta\mathbf{m}$ of the current state towards the optimal value $\hat{\mathbf{m}}$.

The cost function $F(\mathbf{m})$ can be approximated by its second-order Taylor series expansion $\hat{F}(\mathbf{m})$ in the vicinity of its actual state $\mathbf{m}_k$:

$$\begin{aligned}
\mathrm{F}(\mathbf{m}_k+\Delta\mathbf{m}) &\approx \hat{F}_k(\mathbf{m}_k+\Delta\mathbf{m}) \\
&= \mathrm{F}(\mathbf{m}_k)+\underbrace{\left.\frac{\partial \mathrm{F}}{\partial \mathbf{m}}\right|_{\mathbf{m}=\mathbf{m}_k}}_{\nabla_{\mathbf{m}}\mathrm{F}(\mathbf{m}_k)}\Delta\mathbf{m}+\frac{1}{2}\Delta\mathbf{m}^T\underbrace{\left.\frac{\partial^2 \mathrm{F}}{\partial \mathbf{m}\partial \mathbf{m}^T}\right|_{\mathbf{m}=\mathbf{m}_k}}_{\nabla^2_{\mathbf{m}}\mathrm{F}(\mathbf{m}_k)}\Delta\mathbf{m} \\
&= \mathrm{F}(\mathbf{m}_k)+\underbrace{\nabla_{\mathbf{m}}\mathrm{F}(\mathbf{m}_k)}_{\mathbf{g}_k^T}\Delta\mathbf{m}+\frac{1}{2}\Delta\mathbf{m}^T\underbrace{\nabla^2_{\mathbf{m}}\mathrm{F}(\mathbf{m}_k)}_{\mathbf{H}_k}\Delta\mathbf{m} \\
&= \mathrm{F}(\mathbf{m}_k)+\mathbf{g}_k^T\Delta\mathbf{m}+\frac{1}{2}\Delta\mathbf{m}^T\mathbf{H}_k\Delta\mathbf{m} \qquad \text{(A.10)}
\end{aligned}$$

where we introduce the first and second-order derivatives of $F(\mathbf{m})$, namely the gradient vector $\mathbf{g_k} = \nabla_{\mathbf{m}}\mathrm{F}(\mathbf{m})^\top$ and the Hessian matrix $\mathbf{H_k} = \nabla^2_{\mathbf{m}}\mathrm{F}(\mathbf{m_k})$.

Taking now derivatives with respect to an increment in the unknowns, we obtain

$$\begin{aligned}
\frac{\partial \mathrm{F}(\mathbf{m}_k+\Delta\mathbf{m})}{\partial \Delta\mathbf{m}} &\approx \frac{\partial \hat{F}_k(\mathbf{m}_k+\Delta\mathbf{m})}{\partial \Delta\mathbf{m}} \qquad \text{(A.11)} \\
&= \underbrace{\frac{\partial}{\partial \Delta\mathbf{m}}\{\mathrm{F}(\mathbf{m}_k)\}}_{0}+\frac{\partial}{\partial \Delta\mathbf{m}}\{\mathbf{g}^T\Delta\mathbf{m}\}+\frac{\partial}{\partial \Delta\mathbf{m}}\left\{\frac{1}{2}\Delta\mathbf{m}^T\mathbf{H}\Delta\mathbf{m}\right\}
\end{aligned}$$

Provided that $\frac{\partial \mathbf{a^T M a}}{\partial \mathbf{a}} = \left(\mathbf{M}+\mathbf{M^T}\right)\mathbf{a}$ and $\frac{\partial \mathbf{a^T b}}{\partial \mathbf{b}} = \mathbf{a}$, and since the Hessian matrix is symmetric:

$$\frac{\partial \hat{F}_k(\mathbf{m}_k+\Delta\mathbf{m})}{\partial \Delta\mathbf{m}} = \mathbf{g}+\mathbf{H}\Delta\mathbf{m} \qquad \text{(A.12)}$$

The increment of the current state that leads to the optimal value is then determined from

$$\left.\frac{\partial \hat{F}_k(\mathbf{m}_k+\Delta\mathbf{m})}{\partial \Delta\mathbf{m}}\right|_{\Delta\mathbf{m}=0} = 0 \;\rightarrow\; \mathbf{g}_k+\mathbf{H}_k\Delta\mathbf{m}_k = 0$$

Therefore, $\Delta\mathbf{m}_k$ is computed by solving the linear system of the form $\mathbf{Ax} = \mathbf{b}$

$$\mathbf{H_k}\Delta\mathbf{m_k} = -\mathbf{g_k} \qquad \text{(A.13)}$$

This linear system can be rewritten by taking the first order Taylor extension on the residuals $\mathbf{r}$, leading to the well-known formula:

$$\underbrace{(\mathbf{J}^\top\Lambda\mathbf{J})}_{\text{Hessian } \mathbf{H}}\Delta\mathbf{m} = -\underbrace{\mathbf{J}^\top\Lambda\mathbf{r}}_{\text{Gradient } \mathbf{g}} \qquad \text{(A.14)}$$

where $\mathbf{J}$ stands for the Jacobian of the residuals with respect to $\Delta\mathbf{m}$

$$\mathbf{J} = \frac{\partial \mathbf{r}\left(\mathbf{m}_k + \Delta\mathbf{m}\right)}{\partial \Delta\mathbf{m}} \tag{A.15}$$

Appendix B
Lie algebra and Lie groups

A Lie group is defined as a smooth differentiable manifold for which the group axioms apply. That is, there is an operator that combines two elements of the group into a third element also in the same group, which fulfils the axioms of associativity, identity and invertibility. A Lie algebra can be described as a representation of a Lie group in a vector space where infinitesimal transformations can be applied.

The special orthogonal group $\mathbb{SO}(3)$ which represents all rotations in 3D Euclidean space $\mathbb{R}^3$, is an example of a Lie group. Each rotation in $\mathbb{SO}(3)$ is expressed as a 3×3 orthonormal matrix. The Lie algebra corresponding to the Lie group $\mathbb{SO}(3)$ is expressed as $\mathfrak{so}(3)$, and coincides with $\mathbb{R}^3$. It constitutes a minimal parameterization for the rotations which can be arithmetically manipulated as a vector space. The transformation from the Lie algebra $\mathfrak{so}(3)$ to its corresponding Lie group $\mathbb{SO}(3)$ is defined by the exponential map operation

$$\begin{array}{c} \exp\colon \mathfrak{so}(3) \rightarrow \mathbb{SO}(3) \\ \omega \quad \rightarrow \quad \mathrm{R} \end{array}$$

which is given by the Rodrigues' formula

$$\mathrm{R} = I + \frac{\sin(\theta)}{\theta}[\omega]_\times + \frac{(1-\cos(\theta))}{\theta^2}[\omega]_\times^2 \tag{B.1}$$

where $\theta = |\omega|$, and $[\cdot]_\times$ represents the skew-symmetric matrix operator. The inverse operation is called logarithm map, which in this case is obtained from

$$\theta = \arccos\left(\frac{\mathrm{trace}(R)-1}{2}\right) \tag{B.2}$$

$$\omega = \begin{cases} 0 & \text{if } \theta = 0 \\ \frac{\theta}{2\sin(\theta)}(R - R^\top) & \text{if } \theta \neq 0 \text{ and } \theta \in (-\pi, \pi) \end{cases} \tag{B.3}$$

An intuitive representation for the smooth manifold of a Lie group (e.g. $\mathbb{SO}(3)$) and the Euclidean tangent space of its Lie algebra ,showing the transformations between

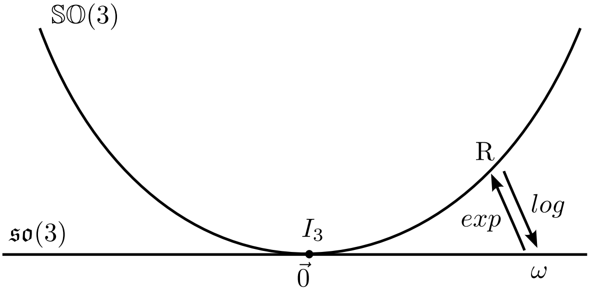


Figure B.1: Lie group and Lie algebra for the space of 3D rotations.

both, is depicted in figure B.1. We can see that the tangent point is located at the identity of the Lie group.

Another interesting Lie group which is used throughout this thesis is the special Euclidean group $\mathbb{SE}(3)$ representing 3D rigid motions (rotation plus translation). An element of this group is expressed as a 4 × 4 matrix

$$\mathrm{T} = [\mathrm{R}|\mathbf{t}] = \left[\begin{array}{c|c} \mathrm{R} & \mathbf{t} \\ \hline 0\ 0\ 0 & 1 \end{array}\right] \tag{B.4}$$

where the rotation $\mathrm{R} \in \mathbb{SO}(3)$ and the translations $\mathbf{t} \in \mathbb{R}^3$. In this case, the exponential map is formulated as

$$\begin{aligned} \exp\colon \mathfrak{se}(3) &\to \mathbb{SE}(3) \\ \begin{pmatrix} \omega \\ t' \end{pmatrix} &\to \left[\begin{array}{c|c} \mathrm{R} & \mathbf{t} \\ \hline 0\ 0\ 0 & 1 \end{array}\right] \end{aligned}$$

where the rotation matrix R is calculated as explained above in B.1, and the translation is given by

$$\mathbf{t} = \mathrm{R}\mathbf{t}' \tag{B.5}$$

The logarithm map can be trivially derived from the above formulation.

The above transformations between Lie groups and Lie algebras are required by many optimization algorithms in the field of robotics and computer vision, where rotation or pose parameters need to be estimated to compute robot or camera motion. The differentiation of a function which depends on some parameters of the Lie group is always done at the tangent point in the Euclidean space defined by its Lie algebra. Thus, direct operations like the calculation of residuals are calculated with the Lie group formulation, while the optimization problem is derived in the tangent space of the manifold given by the Lie algebra. For a more detailed description of Lie groups and Lie algebras in the context of mobile robotics, the reader is referred to [Blanco, 2010].

Appendix C
Propagation of uncertainty

This appendix details the propagation of uncertainty for Maximum Likelihood Estimation (MLE) problems. This permits taking into account the uncertainty of the sensor measurements in the minimization of the cost function. For that, the covariance of different functions must be estimated. This is systematically done through linearization, so that the covariance Σ_y of

$$y = f(x) \tag{C.1}$$

is computed as

$$\Sigma_y = J_x \Sigma_x J_x^T \tag{C.2}$$

being $J_x = \frac{\partial f(x)}{\partial x}$ the Jacobian of $f(x)$. This is an approximation when $f(x)$ is not linear, otherwise the formula is exact.

As an example, the variances of the error functions used in 2.3.1.3 are derived here. Thus, to compute the variance of the error in eq. 2.15

$$r_{jk} = \underbrace{R_j \mathbf{l}_j \times R_k \mathbf{l}_k}_{n_{jk}} \cdot \underbrace{(R_j \mathbf{c}_j + \mathbf{t_j} - \hat{R}_k \mathbf{c}_k - \mathbf{t_k})}_{d}$$

the linearization above is applied, resulting in

$$\sigma^2 \simeq \mathbf{n}_{jk}^T \sigma_d \mathbf{n}_{jk} + \mathbf{d}^T \Sigma_{n_{jk}} \mathbf{d} \tag{C.3}$$

$$\Sigma_{n_{jk}} = [R_j \mathbf{l}_j]_x R_k \Sigma_{l_k} R_k^T [R_j \mathbf{l}_j]_x^T + [R_k \mathbf{l}_k]_x R_j \Sigma_{l_j} R_j^T [R_k \mathbf{l}_k]_x^T \tag{C.4}$$

$$\sigma_d^2 = R_j \Sigma_{c_j} R_j^T + [\mathbf{c}_j]_x \Sigma_{R_j} [\mathbf{c}_j]_x^T + R_k \Sigma_{c_k} R_k^T + [\mathbf{c}_k]_x \Sigma_{R_k} [\mathbf{c}_k]_x^T \tag{C.5}$$

where $[\cdot]_\times$ is the skew-symmetric matrix operator.

For the case of the error in eq. 2.17,

$$r_i = \mathbf{n}_{jk}^a \cdot \mathbf{n}_{jk}^b \tag{C.6}$$

the linearization above is applied as

$$\sigma^2 = (\mathbf{n}^a_{jk})^T R_j^T R_k \Sigma_{\mathbf{n}^b_{jk}} R_k^T R_j \mathbf{n}^a_{jk} + (\mathbf{n}^b_{jk})^T R_k^T R_j \Sigma_{\mathbf{n}^a_{jk}} R_j^T R_k \mathbf{n}^b_{jk} \tag{C.7}$$

These linearisations imply that the product of two Gaussian random variables is approximated as a Gaussian distribution, despite that the result follows a χ^2 distribution. It can be verified that the error of this approximation diminishes asymptotically with the number of samples (observations) [Severo and Zelen, 1960; Zhang, 2005].

Appendix D
Fisher Information and Cramér-Rao Bound

The Fisher Information Matrix (FIM) is a statistic measure of how much information an observable random variable X carries about the parameters $\theta = [\theta_1, \theta_2, \ldots, \theta_N]^T$ upon which the likelihood function $P(X|\theta)$ depends. For an unbiased estimator, the FIM is defined as

$$\mathscr{I}(\theta) = \mathrm{E}\left[\,(\nabla_\theta \log P(X|\theta))\,(\nabla_\theta \log P(X|\theta))\,|\,\theta\right] \tag{D.1}$$

which is a $N \times N$ positive semidefinite symmetric matrix. Formally, the FIM corresponds to the expected value of the observed information. It has important implications regarding the observability of estimation problems, concretely, the problem is observable (it has a solution) if and only if the FIM has full rank, i.e. $rank(\mathscr{I}) = N$.

Considering an estimation problem in which the observable random variables X follow an unbiased, asymptotically Gaussian distribution, the FIM can be calculated as

$$\mathscr{I}(\theta) = J_E^T \Sigma_X^{-1} J_E \tag{D.2}$$

where J_E is the Jacobian of the observation equations (i.e. the cost function of the estimation problem) and Σ_X is the covariance of the observed data X [Van Trees and Bell, 2007]. This formulation has a relevant significance implying that, for of a maximum likelihood estimation problem which is solved through a least squares minimization of the negative log-likelihood, the FIM corresponds to the Hessian of such negative log-likelihood. This result is applied in the observability analysis of all the extrinsic calibration problems tackled in chapter 2.

The Cramér-Rao Bound (CRB) defines a lower bound for the variance of estimators, therefore, it is a measure of the performance of estimators [Van Trees and Bell, 2007]. The CRB states that the covariance of an unbiased estimator cannot be lower than the inverse of the Fisher information [Van Trees and Bell, 2007].

$$\mathrm{cov}(\hat{\theta}) \geq \mathscr{I}^{-1}(\theta) \tag{D.3}$$

where $\hat{\theta}$ is the true value of the estimator. If this limit is achieved, the estimator is called efficient. Thus, the FIM defines a lower bound for the noise of our estimate, which can be used to find the best estimator for the problem.

Bibliography

[Angeli *et al.*, 2009] Angeli, A., Doncieux, S., Meyer, J.-A., and Filliat, D. (2009). Visual topological slam and global localization. *IEEE International Conference on Robotics and Automation.*

[Angeli *et al.*, 2008] Angeli, A., Filliat, D., Doncieux, S., and Meyer, J. (2008). Fast and incremental method for loop-closure detection using bags of visual words. *Robotics, IEEE Transactions on*, 24(5):1027–1037.

[Argiles *et al.*, 2011] Argiles, A., Civera, J., and Montesano, L. (2011). Dense multi-planar scene estimation from a sparse set of images. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*, pages 4448–4454. IEEE.

[Arras and Siegwart, 1998] Arras, K. O. and Siegwart, R. Y. (1998). Feature extraction and scene interpretation for map-based navigation and map building. In *Intelligent Systems & Advanced Manufacturing*, pages 42–53.

[Arun *et al.*, 1987] Arun, K. S., Huang, T. S., and Blostein, S. D. (1987). Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 9(5):698–700.

[AsusXPL, 2011] AsusXPL (2011). Asus xtion pro live specifications. `http://www.asus.com/es/Multimedia/Xtion_PRO_LIVE/specifications/` (accessed july/2014).

[Bailey and Durrant-Whyte, 2006] Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117.

[Barber *et al.*, 2008] Barber, D., Mills, J., and Smith-Voysey, S. (2008). Geometric validation of a ground-based mobile laser scanning system. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1):128–141.

[Basso *et al.*, 2014a] Basso, F., Levorato, R., and Menegatti, E. (2014a). Online Calibration for Networks of Cameras and Depth Sensors. In *The 12th Workshop on*

Non-classical Cameras, Camera Networks and Omnidirectional Vision (OMNIVIS 2014).

[Basso *et al.*, 2014b] Basso, F., Pretto, A., and Menegatti, E. (2014b). Unsupervised Intrinsic and Extrinsic Calibration of a Camera-Depth Sensor Couple. In *IEEE International Conference on Robotics and Automation (ICRA2014).*

[Bay *et al.*, 2008] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359.

[Besl and McKay, 1992] Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics.

[Bhattacharyya, 1946] Bhattacharyya, A. (1946). On a measure of divergence between two multinomial populations. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 7(4):401–406.

[Biswas and Veloso, 2012] Biswas, J. and Veloso, M. (2012). Depth camera based indoor mobile robot localization and navigation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1697–1702. IEEE.

[Blanco *et al.*, 2013] Blanco, J., González-Jiménez, J., and Fernández-Madrigal, J. (2013). Sparser relative bundle adjustment (srba): constant-time maintenance and local optimization of arbitrarily large maps. In *IEEE International Conference on Robotics and Automation.*

[Blanco, 2008] Blanco, J.-L. (2008). Development of scientific applications with the mobile robot programming toolkit. *The MRPT reference book. Machine Perception and Intelligent Robotics Laboratory, University of Málaga, Málaga, Spain.*

[Blanco, 2009] Blanco, J. L. (2009). Contributions to localization, mapping and navigation in mobile robotics. *Ph.D. dissertation.*

[Blanco, 2010] Blanco, J.-L. (2010). A tutorial on se(3) transformation parameterizations and on-manifold optimization. *University of Malaga, Tech. Rep.*

[Blanco *et al.*, 2008] Blanco, J.-L., Fernández-Madrigal, J.-A., and Gonzalez, J. (2008). Toward a unified bayesian approach to hybrid metric-topological slam. *Robotics, IEEE Transactions on*, 24(2):259–270.

[Blanco *et al.*, 2006] Blanco, J.-L., González, J., and Fernández-Madrigal, J.-A. (2006). Consistent observation grouping for generating metric-topological maps that improves robot localization. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 818–823.

[Blanco *et al.*, 2009a] Blanco, J.-L., González-Jiménez, J., and Fernández-Madrigal, J.-A. (2009a). Subjective local maps for hybrid metric-topological slam. *Robotics and Autonomous Systems*, 57(1):64–74.

[Blanco *et al.*, 2009b] Blanco, J.-L., Moreno, F.-A., and Gonzalez, J. (2009b). A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots*, 27(4):327–351.

[Blanco-Claraco *et al.*, 2014] Blanco-Claraco, J.-L., Moreno-Dueñas, F.-Á., and González-Jiménez, J. (2014). The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. *The International Journal of Robotics Research*, 33(2):207–214.

[Bohren *et al.*, 2008] Bohren, J., Foote, T., Keller, J., Kushleyev, A., Lee, D., Stewart, A., Vernaza, P., Derenick, J., Spletzer, J., and Satterfield, B. (2008). Little ben: The ben franklin racing team's entry in the 2007 darpa urban challenge. *Journal of Field Robotics*, 25(9):598–614.

[Borges and Aldon, 2000] Borges, G. A. and Aldon, M.-J. (2000). A split-and-merge segmentation algorithm for line extraction in 2d range images. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 1, pages 441–444. IEEE.

[Borrmann *et al.*, 2011] Borrmann, D., Elseberg, J., Lingemann, K., and Nüchter, A. (2011). The 3d hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Research*, 2(2):1–13.

[Borrmann *et al.*, 2008] Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A., and Hertzberg, J. (2008). Globally consistent 3d mapping with scan matching. *Robotics and Autonomous Systems*, 56(2):130–142.

[Bosse *et al.*, 2003] Bosse, M., Newman, P., Leonard, J., Soika, M., Feiten, W., and Teller, S. (2003). An atlas framework for scalable mapping. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2:1899–1906.

[Bosse and Zlot, 2008] Bosse, M. and Zlot, R. (2008). Map matching and data association for large-scale two-dimensional laser scan-based slam. *The International Journal of Robotics Research*, 27(6):667–691.

[Bosse and Zlot, 2010] Bosse, M. and Zlot, R. (2010). Place recognition using regional point descriptors for 3d mapping. In *Field and Service Robotics*, pages 195–204. Springer.

[Brookshire and Teller, 2012] Brookshire, J. and Teller, S. J. (2012). Extrinsic calibration from per-sensor egomotion. In *Robotics: Science and Systems*.

[Campbell *et al.*, 2010] Campbell, M., Egerstedt, M., How, J. P., and Murray, R. M. (2010). Autonomous driving in urban environments: approaches, lessons and challenges. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1928):4649–4672.

[Carr *et al.*, 2001] Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. (2001). Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76. ACM.

[Censi *et al.*, 2013] Censi, A., Franchi, A., Marchionni, L., and Oriolo, G. (2013). Simultaneous calibration of odometry and sensor parameters for mobile robots. *Robotics, IEEE Transactions on*, 29(2):475–492.

[Chapoulie *et al.*, 2011] Chapoulie, A., Rives, P., and Filliat, D. (2011). A spherical representation for efficient visual loop closing. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 335–342. IEEE.

[Chekhlov *et al.*, 2007] Chekhlov, D., Gee, A., Calway, A., and Mayol-Cuevas, W. (2007). Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 1–4. IEEE Computer Society.

[Chen and Medioni, 1992] Chen, Y. and Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155.

[Comaniciu and Meer, 1997] Comaniciu, D. and Meer, P. (1997). Robust analysis of feature spaces: color image segmentation. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 750 –755.

[Cowan and Koditschek, 1999] Cowan, N. J. and Koditschek, D. E. (1999). Planar image based visual servoing as a navigation problem. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pages 611–617. IEEE.

[Csurka *et al.*, 2004] Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, page 22.

[Cummins and Newman, 2008] Cummins, M. and Newman, P. (2008). Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665.

[Davison, 2003] Davison, A. J. (2003). Real-time simultaneous localisation and mapping with a single camera. *Proceedings of the International Conference on Computer Vision (ICCV).*

[Davison and Murray, 2002] Davison, A. J. and Murray, D. W. (2002). Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[Dellaert *et al.*, 1999] Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1322–1328. IEEE.

[Devaux *et al.*, 2013] Devaux, J.-C., Hadj-Abdelkader, H., Colle, E., *et al.* (2013). A multi-sensor calibration toolbox for kinect: Application to kinect and laser range finder fusion. In *Proc. of the 16th International Conference on Advanced Robotics (ICAR 2013).*

[Devernay and Faugeras, 1995] Devernay, F. and Faugeras, O. D. (1995). Automatic calibration and removal of distortion from scenes of structured environments. In *SPIE's 1995 International Symposium on Optical Science, Engineering, and Instrumentation*, pages 62–72. International Society for Optics and Photonics.

[Diosi and Kleeman, 2003] Diosi, A. and Kleeman, L. (2003). Uncertainty of line segments extracted from static sick pls laser scans. In *SICK PLS laser. In Australiasian Conference on Robotics and Automation.*

[Dissanayake *et al.*, 2001] Dissanayake, M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., and Csorba, M. (2001). A solution to the simultaneous localization and map building (slam) problem. *Robotics and Automation, IEEE Transactions on*, 17(3):229–241.

[Durrant-Whyte and Bailey, 2006] Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110.

[Eade and Drummond, 2007] Eade, E. and Drummond, T. (2007). Monocular slam as a graph of coalesced observations. In *International Conference on Computer Vision.*

[Elfes, 1989] Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57.

[Estrada *et al.*, 2005] Estrada, C., Neira, J., and Tardos, J. (2005). Hierarchical slam: Real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596.

[Faugeras and Toscani, 1986] Faugeras, O. D. and Toscani, G. (1986). The calibration problem for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 86, pages 15–20.

[Fernández-Madrigal and Claraco, 2013] Fernández-Madrigal, J.-A. and Claraco, J. L. B. (2013). *Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods*. Information Science Reference.

[Fernández-Moral *et al.*, 2014a] Fernández-Moral, E., Arévalo, V., and González-Jiménez, J. (2014a). A compact planar patch descriptor based on color. In *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.

[Fernández-Moral *et al.*, 2015a] Fernández-Moral, E., Arévalo, V., and González-Jiménez, J. (2015a). Hybrid metric-topological mapping for large scale monocular slam. In *Informatics in Control, Automation and Robotics*, volume 325 of *Lecture Notes in Electrical Engineering*, pages 217–232. Springer International Publishing.

[Fernández-Moral *et al.*, 2013a] Fernández-Moral, E., González-Jiménez, J., and Arévalo, V. (2013a). Creating metric-topological maps for large-scale monocular slam. In *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.

[Fernández-Moral *et al.*, 2015b] Fernández-Moral, E., González-Jiménez, J., and Arévalo, V. (2015b). Extrinsic calibration of 2d laser rangefinders from perpendicular plane observations. *The International Journal of Robotics Research*.

[Fernández-Moral *et al.*, 2014b] Fernández-Moral, E., González-Jiménez, J., Rives, P., and Arévalo, V. (2014b). Extrinsic calibration of a set of range cameras in 5 seconds without pattern. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

[Fernández-Moral *et al.*, 2013b] Fernández-Moral, E., Mayol-Cuevas, W., Arévalo, V., and González-Jiménez, J. (2013b). Fast place recognition with plane-based maps. In *International Conference on Robotics and Automation (ICRA)*. IEEE.

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

[Forkuo and King, 2004] Forkuo, E. K. and King, B. (2004). Automatic fusion of photogrammetric imagery and laser scanner point clouds. *International Archives of Photogrammetry and Remote Sensing*, 35:921–926.

[Forsberg *et al.*, 1995] Forsberg, J., Larsson, U., and Wernersson, A. (1995). Mobile robot navigation using the range-weighted hough transform. *Robotics & Automation Magazine, IEEE*, 2(1):18–26.

[Foxlin, 2002] Foxlin, E. M. (2002). Generalized architecture for simultaneous localization, auto-calibration, and map-building. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 527–533. IEEE.

[Furukawa *et al.*, 2009] Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. (2009). Manhattan-world stereo. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1422–1429. IEEE.

[Galindo *et al.*, 2008] Galindo, C., Fernández-Madrigal, J.-A., González, J., and Saffiotti, A. (2008). Robot task planning using semantic maps. *Robotics and Autonomous Systems*, 56(11):955–966.

[Galindo *et al.*, 2005] Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., Fernández-Madrigal, J., and González, J. (2005). Multi-hierarchical semantic maps for mobile robotics. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2278–2283.

[Gao and Spletzer, 2010] Gao, C. and Spletzer, J. R. (2010). On-line calibration of multiple lidars on a mobile vehicle platform. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 279–284. IEEE.

[Gaspar *et al.*, 2000] Gaspar, J., Winters, N., and Santos-Victor, J. (2000). Vision-based navigation and environmental representations with an omnidirectional camera. *Robotics and Automation, IEEE Transactions on*, 16(6):890–898.

[Gee *et al.*, 2008] Gee, A., Chekhlov, D., Calway, A., and Mayol-Cuevas, W. (2008). Discovering higher level structure in visual slam. *Robotics, IEEE Transactions on*, 24(5):980 –990.

[Gevers and Smeulders, 1999] Gevers, T. and Smeulders, W. (1999). Color based object recognition. *Pattern recognition*, 32(3):453–464.

[Glas *et al.*, 2010] Glas, D. F., Miyashita, T., Ishiguro, H., and Hagita, N. (2010). Automatic position calibration and sensor displacement detection for networks of laser range finders for human tracking. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2938–2945. IEEE.

[Glennie and Lichti, 2010] Glennie, C. and Lichti, D. D. (2010). Static calibration and analysis of the velodyne hdl-64e s2 for high accuracy mobile scanning. *Remote Sensing*, 2(6):1610–1624.

[Goedemé *et al.*, 2007] Goedemé, T., Nuttin, M., Tuytelaars, T., and Van Gool, L. (2007). Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, 74(3):219–236.

[Gokhool *et al.*, 2014] Gokhool, T., Meilland, M., Rives, P., and Fernández-Moral, E. (2014). A Dense Map Building Approach from Spherical RGBD Images. In *International Conference on Computer Vision Theory and Applications (VISAPP 2014)*, Lisbon, Portugal.

[Granström *et al.*, 2011] Granström, K., Schön, T. B., Nieto, J. I., and Ramos, F. T. (2011). Learning to close loops from range data. *The international journal of robotics research*, 30(14):1728–1754.

[Grimson, 1990] Grimson, W. E. L. (1990). *Object Recognition by Computer - The role of Geometric Constraints*. MIT Press, Cambridge, MA.

[Grisetti *et al.*, 2010] Grisetti, G., Kummerle, R., Stachniss, C., and Burgard, W. (2010). A tutorial on graph-based slam. *Intelligent Transportation Systems Magazine, IEEE*, 2(4):31–43.

[Guo and Roumeliotis, 2013] Guo, C. and Roumeliotis, S. (2013). Imu-rgbd camera 3d pose estimation and extrinsic calibration: Observability analysis and consistency improvement. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2935–2942.

[Ha, 2012] Ha, J.-E. (2012). Extrinsic calibration of a camera and laser range finder using a new calibration structure of a plane with a triangular hole. *International Journal of Control, Automation and Systems*, 10(6):1240–1244.

[Haala *et al.*, 2008] Haala, N., Peter, M., Kremer, J., and Hunter, G. (2008). Mobile lidar mapping for 3d point cloud collection in urban areas - a performance test. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37:1119–1127.

[Hafner *et al.*, 1995] Hafner, J., Sawhney, H., Equitz, W., Flickner, M., and Niblack, W. (1995). Efficient color histogram indexing for quadratic form distance functions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(7):729 –736.

[Haines *et al.*, 2013] Haines, O., Martınez-Carranza, J., and Calway, A. (2013). Visual mapping using learned structural priors. In *Robotics and Automation (ICRA), in 2013 IEEE International Conference on*. IEEE.

[Hansen *et al.*, 2008] Hansen, P., Mladenović, N., and Pérez, J. A. M. (2008). Variable neighbourhood search: methods and applications. *4OR*, 6(4):319–360.

[Hartley and Zisserman, 2003] Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

[Heikkila and Silvén, 1997] Heikkila, J. and Silvén, O. (1997). A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112. IEEE.

[Heng *et al.*, 2013] Heng, L., Li, B., and Pollefeys, M. (2013). Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In *International Conference on Intelligent Robots and Systems (IROS 2013)*, pages 1793–1800. IEEE/RSJ.

[Herrera *et al.*, 2011] Herrera, D., Kannala, J., and Heikkilä, J. (2011). Accurate and practical calibration of a depth and color camera pair. In *Computer Analysis of Images and Patterns*, pages 437–445. Springer.

[Hirschmuller, 2005] Hirschmuller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE.

[Hoiem *et al.*, 2007] Hoiem, D., Efros, A. A., and Hebert, M. (2007). Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172.

[Holmes *et al.*, 2009] Holmes, S., Sibley, G., Klein, G., and Murray, D. (2009). A relative frame representation for fixed-time bundle adjustment in monocular sfm. In *IEEE International Conference on Robotics and Automation*.

[Holz and Behnke, 2012] Holz, D. and Behnke, S. (2012). Fast range image segmentation and smoothing using approximate surface reconstruction and region growing. *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS), Jeju Island, Korea.*

[Holz and Behnke, 2013] Holz, D. and Behnke, S. (2013). Fast range image segmentation and smoothing using approximate surface reconstruction and region growing. In *Intelligent Autonomous Systems 12*, pages 61–73. Springer.

[Holzer *et al.*, 2012] Holzer, S., Rusu, R. B., Dixon, M., Gedikli, S., and Navab, N. (2012). Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2684–2689. IEEE.

[Hoover *et al.*, 1996] Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., Bowyer, K., Eggert, D. W., Fitzgibbon, A., and Fisher, R. B. (1996). An experimental comparison of range image segmentation algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(7):673–689.

[Huang *et al.*, 2010] Huang, A. S., Antone, M., Olson, E., Fletcher, L., Moore, D., Teller, S., and Leonard, J. (2010). A high-rate, heterogeneous data set from the darpa urban challenge. *The International Journal of Robotics Research*, 29(13):1595–1601.

[Huber, 1981] Huber, P. J. (1981). Robust statistics.

[Jogan and Leonardis, 2000] Jogan, M. and Leonardis, A. (2000). Robust localization using panoramic view-based recognition. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 136–139. IEEE.

[Kerl *et al.*, 2013a] Kerl, C., Sturm, J., and Cremers, D. (2013a). Dense visual slam for rgb-d cameras. In *International Conference on Intelligent Robots and Systems (IROS 2013)*. IEEE/RSJ.

[Kerl *et al.*, 2013b] Kerl, C., Sturm, J., and Cremers, D. (2013b). Robust odometry estimation for rgb-d cameras. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*.

[Kim and Chung, 2003] Kim, J.-H. and Chung, M. J. (2003). Slam with omnidirectional stereo vision sensor. In *International Conference on Intelligent Robots and Systems (IROS 2003)*, volume 1, pages 442–447. IEEE/RSJ.

[Klank *et al.*, 2009] Klank, U., Pangercic, D., Rusu, R. B., and Beetz, M. (2009). Real-time cad model matching for mobile manipulation and grasping. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 290–296. IEEE.

[Klein and Murray, 2007] Klein, G. and Murray, D. W. (2007). Parallel tracking and mapping for small ar workspaces. *In Proceedings of the International Symposium on Mixed and Augmented Reality*.

[Konolige, 2010] Konolige, K. (2010). Sparse sparse bundle adjustment. In *British Machine Vision Conference*.

[Konolige and Bowman, 2009] Konolige, K. and Bowman, J. (2009). Towards lifelong visual maps. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1156–1163. IEEE.

[Koppula *et al.*, 2011] Koppula, H. S., Anand, A., Joachims, T., and Saxena, A. (2011). Semantic labeling of 3d point clouds for indoor scenes. In *Advances in Neural Information Processing Systems*, pages 244–252.

[Košecká and Zhang, 2005] Košecká, J. and Zhang, W. (2005). Extraction, matching, and pose recovery based on dominant rectangular structures. *Computer Vision and Image Understanding*, 100(3):274–293.

[Kröse *et al.*, 2001] Kröse, B. J., Vlassis, N., Bunschoten, R., and Motomura, Y. (2001). A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, 19(6):381–391.

[Kummerle *et al.*, 2011] Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). g 2 o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613. IEEE.

[Lafarge and Alliez, 2013] Lafarge, F. and Alliez, P. (2013). Surface reconstruction through point set structuring. In *Proc. of Eurographics*, Girona, Spain.

[Larsen *et al.*, 1998] Larsen, T. D., Bak, M., Andersen, N. A., and Ravn, O. (1998). Location estimation for an autonomously guided vehicle using an augmented kalman filter to autocalibrate the odometry. In *FUSION98 Spie Conference*. Citeseer.

[Lazebnik *et al.*, 2006] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE.

[Le and Ng, 2009] Le, Q. V. and Ng, A. Y. (2009). Joint calibration of multiple sensors. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3651–3658. IEEE.

[Leonard *et al.*, 2008] Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., Fletcher, L., Frazzoli, E., Huang, A., Karaman, S., *et al.* (2008). A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25(10):727–774.

[Li *et al.*, 2007] Li, G., Liu, Y., Dong, L., Cai, X., and Zhou, D. (2007). An algorithm for extrinsic parameters calibration of a camera and a laser range finder using line features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, pages 3854–3859. IEEE.

[Lieberknecht *et al.*, 2011] Lieberknecht, S., Huber, A., Ilic, S., and Benhimane, S. (2011). Rgb-d camera-based parallel tracking and meshing. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 147–155. IEEE.

[Lim *et al.*, 2011] Lim, J., Pollefeys, M., and Frahm, J.-M. (2011). Online environment mapping. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[Lozano Albalate *et al.*, 2002] Lozano Albalate, M., Devy, M., Miguel, J., and Marti, S. (2002). Perception planning for an exploration task of a 3d environment. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 704–707. IEEE.

[Macknojia *et al.*, 2013] Macknojia, R., Chávez-Aragón, A., Payeur, P., and Laganière, R. (2013). Calibration of a network of kinect sensors for robotic inspection over a large workspace. In *Robot Vision (WORV), 2013 IEEE Workshop on*, pages 184–190. IEEE.

[Manjunath *et al.*, 2001] Manjunath, B., Ohm, J.-R., Vasudevan, V., and Yamada, A. (2001). Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):703 –715.

[Martinelli, 2011] Martinelli, A. (2011). State estimation based on the concept of continuous symmetry and observability analysis: The case of calibration. *Robotics, IEEE Transactions on*, 27(2):239–255.

[Martinelli *et al.*, 2007] Martinelli, A., Tomatis, N., and Siegwart, R. (2007). Simultaneous localization and odometry self calibration for mobile robot. *Autonomous Robots*, 22(1):75–85.

[Martínez-Carranza and Calway, 2010] Martínez-Carranza, J. and Calway, A. (2010). Unifying planar and point mapping in monocular slam. *British Machine Vision Conference*, pages 1–11.

[Martinez-Carranza and Calway, 2012] Martinez-Carranza, J. and Calway, A. (2012). Efficient visual odometry using a structure-driven temporal map. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 5210–5215. IEEE.

[Meilland *et al.*, 2010] Meilland, M., Comport, A. I., and Rives, P. (2010). A spherical robot-centered representation for urban navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, pages 5196–5201. IEEE.

[Meilland *et al.*, 2011] Meilland, M., Comport, A. I., and Rives, P. (2011). Dense visual mapping of large scale environments for real-time localisation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4242–4248. IEEE.

[Menegatti *et al.*, 2002] Menegatti, E., Pagello, E., and Wright, M. (2002). Using omnidirectional vision within the spatial semantic hierarchy. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 908–914. IEEE.

[Menegatti *et al.*, 2006] Menegatti, E., Pretto, A., Scarpa, A., and Pagello, E. (2006). Omnidirectional vision scan matching for robot localization in dynamic environments. *Robotics, IEEE Transactions on*, 22(3):523–535.

[Micušık *et al.*, 2003] Micušık, B., Martinec, D., and Pajdla, T. (2003). 3d metric reconstruction from uncalibrated omnidirectional images. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, volume 1, pages 545–550.

[Micusik *et al.*, 2008] Micusik, B., Wildenauer, H., and Vincze, M. (2008). Towards detection of orthogonal planes in monocular images of indoor environments. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 999–1004. IEEE.

[Miller *et al.*, 2011] Miller, I., Campbell, M., and Huttenlocher, D. (2011). Efficient unbiased tracking of multiple dynamic obstacles under large viewpoint changes. *Robotics, IEEE Transactions on*, 27(1):29–46.

[Mirzaei *et al.*, 2012] Mirzaei, F. M., Kottas, D. G., and Roumeliotis, S. I. (2012). 3d lidar–camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization. *The International Journal of Robotics Research*, 31(4):452–467.

[Mirzaei and Roumeliotis, 2008] Mirzaei, F. M. and Roumeliotis, S. I. (2008). A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation. *Robotics, IEEE Transactions on*, 24(5):1143–1156.

[Moosmann and Stiller, 2011] Moosmann, F. and Stiller, C. (2011). Velodyne slam. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 393–398. IEEE.

[Moravec and Elfes, 1985] Moravec, H. P. and Elfes, A. (1985). High resolution maps from wide angle sonar. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 116–121. IEEE.

[Moreno *et al.*, 2013] Moreno, F.-A., Gonzalez-Jimenez, J., Blanco, J.-L., and Esteban, A. (2013). An instrumented vehicle for efficient and accurate 3d mapping of roads. *Computer-Aided Civil and Infrastructure Engineering*, 28(6):403–419.

[Murase and Nayar, 1995] Murase, H. and Nayar, S. K. (1995). Visual learning and recognition of 3-d objects from appearance. *International journal of computer vision*, 14(1):5–24.

[Nguyen *et al.*, 2005] Nguyen, V., Martinelli, A., Tomatis, N., and Siegwart, R. (2005). A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1929–1934. IEEE.

[Ni and Dellaert, 2010] Ni, K. and Dellaert, F. (2010). Multi-level submap based slam using nested dissection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[Ni *et al.*, 2007] Ni, K., Steedly, D., and Dellaert, F. (2007). Tectonic sam: Exact, out-of-core, submap-based slam. In *IEEE International Conference on Robotics and Automation*.

[Nistér *et al.*, 2004] Nistér, D., Naroditsky, O., and Bergen, J. R. (2004). Visual odometry. *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, pages 652–659.

[Oliva and Torralba, 2001] Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175.

[Oliva and Torralba, 2006] Oliva, A. and Torralba, A. (2006). Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23.

[Pandey *et al.*, 2010] Pandey, G., McBride, J., Savarese, S., and Eustice, R. (2010). Extrinsic calibration of a 3d laser scanner and an omnidirectional camera. In *7th IFAC symposium on intelligent autonomous vehicles*, volume 7.

[Pathak *et al.*, 2010a] Pathak, K., Birk, A., Vaskevicius, N., Pfingsthorn, M., Schwertfeger, S., and Poppinga, J. (2010a). Online three-dimensional slam by registration of large planar surface segments and closed-form pose-graph relaxation. *Journal of Field Robotics*, 27(1):52–84.

[Pathak *et al.*, 2010b] Pathak, K., Birk, A., Vaskevicius, N., and Poppinga, J. (2010b). Fast registration based on noisy planes with unknown correspondences for 3-d mapping. *IEEE Transactions on Robotics*, 26(3):424–441.

[Pathak *et al.*, 2010c] Pathak, K., Vaskevicius, N., and Birk, A. (2010c). Uncertainty analysis for optimum plane extraction from noisy 3d range-sensor point-clouds. *Intelligent Service Robotics*, 3(1):37–48.

[Pathak *et al.*, 2012] Pathak, K., Vaskevicius, N., Bungiu, F., and Birk, A. (2012). Utilizing color information in 3d scan-registration using planar-patches matching. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, pages 371–376.

[Pele and Werman, 2010] Pele, O. and Werman, M. (2010). The quadratic-chi histogram distance family. *Computer Vision–ECCV 2010*, pages 749–762.

[Petrovskaya and Thrun, 2009] Petrovskaya, A. and Thrun, S. (2009). Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots*, 26(2-3):123–139.

[Poppinga *et al.*, 2008] Poppinga, J., Vaskevicius, N., Birk, A., and Pathak, K. (2008). Fast plane detection and polygonalization in noisy 3d range images. In *International Conference on Intelligent Robots and Systems (IROS 2008)*, pages 3378–3383. IEEE/RSJ.

[Prasad *et al.*, 2011] Prasad, D., Quek, C., Leung, M. K. H., and Cho, S.-Y. (2011). A parameter independent line fitting method. In *Pattern Recognition (ACPR), 2011 First Asian Conference on*, pages 441–445. IEEE.

[Rituerto *et al.*, 2012] Rituerto, A., Murillo, A., and Guerrero, J. (2012). Semantic labeling for indoor topological mapping using a wearable catadioptric system. *Robotics and Autonomous Systems*.

[Rituerto *et al.*, 2010] Rituerto, A., Puig, L., and Guerrero, J. J. (2010). Visual slam with an omnidirectional camera. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 348–351. IEEE.

[Rogers and Christensen, 2009] Rogers, J. G. and Christensen, H. I. (2009). Normalized graph cuts for visual slam. *IEEE/RSJ International Conference on Intelligent Robots and Systems.*

[Ruiz-Sarmiento *et al.*, 2014] Ruiz-Sarmiento, J. R., Galindo, C., and González-Jiménez, J. (2014). Mobile robot object recognition through the synergy of probabilistic graphical models and semantic knowledge. In *European Conf. on Artificial Intelligence. Workshop on Cognitive Robotics.*

[Rusu and Cousins, 2011] Rusu, R. B. and Cousins, S. (2011). 3d is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE.

[Rusu *et al.*, 2008] Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., and Beetz, M. (2008). Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941.

[Savelli and Kuipers, 2004] Savelli, F. and Kuipers, B. (2004). Loop-closing and planarity in topological map-building. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1511–1517.

[Saxena *et al.*, 2009] Saxena, A., Sun, M., and Ng, A. Y. (2009). Make3d: Learning 3d scene structure from a single still image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):824–840.

[Scaramuzza *et al.*, 2010] Scaramuzza, D., Fraundorfer, F., and Pollefeys, M. (2010). Closing the loop in appearance-guided omnidirectional visual odometry by using vocabulary trees. *Robotics and Autonomous Systems*, 58(6):820–827.

[Scaramuzza and Siegwart, 2008] Scaramuzza, D. and Siegwart, R. (2008). Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *Robotics, IEEE Transactions on*, 24(5):1015–1026.

[Schenk *et al.*, 2012] Schenk, K., Kolarow, A., Eisenbach, M., Debes, K., and Gross, H.-M. (2012). Automatic calibration of a stationary network of laser range finders by matching movement trajectories. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, pages 431–437. IEEE.

[Schneider *et al.*, 2013] Schneider, S., Luettel, T., and Wuensche, H.-J. (2013). Odometry-based online extrinsic sensor calibration. In *International Conference onIntelligent Robots and Systems (IROS 2013)*, pages 1287–1292. IEEE/RSJ.

[Schwarz and Behnke, 2014] Schwarz, M. and Behnke, S. (2014). Local navigation in rough terrain using omnidirectional height. In *Proceedings of the German conference on robotics (ROBOTIK).*

[Se *et al.*, 2002] Se, S., Lowe, D., and Little, J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *The international Journal of robotics Research*, 21(8):735–758.

[Severo and Zelen, 1960] Severo, N. C. and Zelen, M. (1960). Normal approximation to the chi-square and non-central f probability functions. *Biometrika*, 47(3-4):411–416.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

[Sibley *et al.*, 2009] Sibley, G., Mei, C., Reid, I., , and Newman, P. (2009). Adaptive relative bundle adjustment. *In Robotics Science and Systems Conference*.

[Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE.

[Smisek *et al.*, 2013] Smisek, J., Jancosek, M., and Pajdla, T. (2013). 3d with kinect. In *Consumer Depth Cameras for Computer Vision*, pages 3–25. Springer.

[Sorkine, 2009] Sorkine, O. (2009). Least-squares rigid motion using svd. *Technical notes*, 120.

[Steinbrücker *et al.*, 2011] Steinbrücker, F., Sturm, J., and Cremers, D. (2011). Real-time visual odometry from dense rgb-d images. In *Workshop on Live Dense Reconstruction with Moving Cameras at the Int. Conf. on Computer Vision (ICCV)*, pages 5210–5215. IEEE.

[Strasdat *et al.*, 2011] Strasdat, H., Davison, A., Montiel, J., , and Konolige, K. (2011). Double window optimisation for constant time visual slam. *IEEE International Conference on Computer Vision (ICCV).*

[Strasdat *et al.*, 2010] Strasdat, H., Montiel, J. M. M., and Davison, A. J. (2010). Scale drift-aware large scale monocular slam. *Robotics: Science and Systems.*

[Swain and Ballard, 1991] Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International journal of computer vision*, 7(1):11–32.

[Szeliski and Shum, 1997] Szeliski, R. and Shum, H.-Y. (1997). Creating full view panoramic image mosaics and environment maps. In *International Conference on Computer Graphics and Interactive Techniques*, pages 251–258. ACM Press/Addison-Wesley Publishing Co.

[Tamimi *et al.*, 2006] Tamimi, H., Andreasson, H., Treptow, A., Duckett, T., and Zell, A. (2006). Localization of mobile robots with omnidirectional vision using particle filter and iterative sift. *Robotics and Autonomous Systems*, 54(9):758–765.

[Tardif *et al.*, 2008] Tardif, J.-P., Pavlidis, Y., and Daniilidis, K. (2008). Monocular visual odometry in urban environments using an omnidirectional camera. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2531–2538. IEEE.

[Teichman *et al.*, 2013] Teichman, A., Miller, S., and Thrun, S. (2013). Unsupervised intrinsic calibration of depth sensors via slam. In *Robotics: Science and Systems*, Berlin, Germany.

[Thrun, 1998] Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71.

[Thrun *et al.*, 2000] Thrun, S., Burgard, W., and Fox, D. (2000). A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 321–328. IEEE.

[Thrun *et al.*, 2006] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., *et al.* (2006). Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692.

[Torralba *et al.*, 2003] Torralba, A., Murphy, K. P., Freeman, W. T., and Rubin, M. A. (2003). Context-based vision system for place and object recognition. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 273–280. IEEE.

[Trevor *et al.*, 2012] Trevor, A. J., Rogers, J., and Christensen, H. I. (2012). Planar surface slam with 3D and 2D sensors. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3041–3048. IEEE.

[Ulrich and Nourbakhsh, 2000] Ulrich, I. and Nourbakhsh, I. (2000). Appearance-based place recognition for topological localization. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1023–1029. IEEE.

[Van Trees and Bell, 2007] Van Trees, H. L. and Bell, K. L. (2007). Bayesian bounds for parameter estimation and nonlinear filtering/tracking. *AMC*, 10:12.

[Vasconcelos *et al.*, 2012] Vasconcelos, F., Barreto, J. P., and Nunes, U. (2012). A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2097–2107.

[Victorino *et al.*, 2003] Victorino, A. C., Rives, P., and Borrelly, J.-J. (2003). Safe navigation for indoor mobile robots. part i: a sensor-based navigation framework. *The International Journal of Robotics Research*, 22(12):1005–1118.

[Vosselman *et al.*, 2004] Vosselman, G., Gorte, B. G., Sithole, G., and Rabbani, T. (2004). Recognising structure in laser scanner point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 46(8):33–38.

[Weingarten and Siegwart, 2006] Weingarten, J. and Siegwart, R. (2006). 3D SLAM using planar segments. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3062–3067.

[Weiss, 2006] Weiss, B. (2006). Fast median and bilateral filtering. *ACM Transactions on Graphics (TOG)*, 25(3):519–526.

[Whitehouse and Culler, 2003] Whitehouse, K. and Culler, D. (2003). Macro-calibration in sensor/actuator networks. *Mobile Networks and Applications*, 8(4):463–472.

[Wurm *et al.*, 2010] Wurm, K. M., Hornung, A., Bennewitz, M., Stachniss, C., and Burgard, W. (2010). Octomap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, volume 2.

[Zhang, 2005] Zhang, J.-T. (2005). Approximate and asymptotic distributions of chi-squared–type mixtures with applications. *Journal of the American Statistical Association*, 100(469):273–285.

[Zhang and Pless, 2004] Zhang, Q. and Pless, R. (2004). Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 3, pages 2301–2306. IEEE/RSJ.

[Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334.

[Zhou, 2014] Zhou, L. (2014). A new minimal solution for the extrinsic calibration of a 2d lidar and a camera using three plane-line correspondences. *Sensors Journal, IEEE*, 14(2):442–454.

[Zhou and Deng, 2012] Zhou, L. and Deng, Z. (2012). Extrinsic calibration of a camera and a lidar based on decoupling the rotation from the translation. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 642–648. IEEE.

[Zivkovic *et al.*, 2005] Zivkovic, Z., Bakker, B., and Krose, B. (2005). Hierarchical map building using visual landmarks and geometric constraints. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2480–2485.

[Zuliani *et al.*, 2005] Zuliani, M., Kenney, C., and Manjunath, B. (2005). The multiransac algorithm and its application to detect planar homographies. *IEEE International Conference on Image Processing*.