

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE ELECTRÓNICA



“Localización de Robots Móviles en Espacios Inteligentes
utilizando Cámaras Externas y Marcas Naturales”

TESIS DOCTORAL

Autor

Daniel Pizarro Pérez

Directores

Manuel Mazo Quintas y Enrique Santiso Gómez

2008

A mis padres, hermana y novia, que sufrieron mis ausencias,
a mis amigos cercanos que nunca dejaron de serlo,
a todos ellos que han estado siempre a mi lado,
gracias por hacer esto más fácil.

Agradecimientos

Esta tesis ha implicado mucho tiempo de trabajo y muchas personas que directa o indirectamente han influido en ella y han determinado su finalización. Espero al menos no olvidarme de las importantes.

En primer lugar quiero agradecer a mis tutores Manuel Mazo y Enrique Santiso por sus consejos, apoyo y consideración.

Por otro lado quiero agradecer a muchos compañeros de trabajo por su colaboración directa en la realización de la tesis. Quiero agradecer el trabajo realizado por Dani Peón, Amaia Santiago, Victor Espejo, Javier Rodríguez y David Jiménez, gracias a los cuales tenemos un “Espacio Inteligente” funcional y con los que he disfrutado mucho trabajando. Gracias a Ignacio Fernández por su inestimable ayuda y a Marta Marrón por animarme a seguir adelante con su trabajo, apoyo y colaboración. Muchas gracias a Emilio Bueno por sus consejos, a Álvaro Hernández por hacer que Clermont-Ferrand fuera menos gris y al resto de compañeros del pasillo por su ayuda.

“I would like to thank the people at Hashimoto Lab, specially Hideki Hashimoto, for his advice and help and also Primoz for his support. (minna doumo arigatou gozaimasu!)”.

Tengo mucho que agradecer a Santiago Cóbreces por su comprensión y paciencia y a mi novia Txus por su ánimo y apoyo en todo momento. También tengo que agradecer enormemente a mis padres y hermana por la paciencia, ayuda y consideración.

Por último me gustaría agradecer el apoyo recibido del resto de compañeros del Departamento de Electrónica de la Universidad de Alcalá por su ayuda a todos los niveles, tanto en lo personal como en lo científico.

A todos ellos, muchas gracias por hacer posible este trabajo.

Resumen

En el presente trabajo de tesis se aborda el problema de la obtención de la pose de un robot móvil mediante cámaras estáticas ubicadas en el entorno. El enfoque presentado se basa en el concepto de “Espacio Inteligente”, en el que un sistema distribuido de procesamiento controla las cámaras y el robot. Los antecedentes de posicionamiento de robots mediante cámaras inciden en el uso de balizamiento artificial de los robots. En esta tesis se propone un enfoque que minimiza el conocimiento previo necesario del robot y hace uso de su apariencia natural para obtener la pose.

Se propone en esta tesis un algoritmo de localización basado en la obtención de marcas naturales, detectadas en el plano imagen del conjunto de cámaras, las cuales se corresponden con un modelo tridimensional del robot.

El sistema de localización planteado se divide en dos fases. Primero se obtiene el modelo tridimensional del robot y su pose inicial mediante una fase de inicialización. En segundo lugar, se obtiene la pose del robot en cada instante de tiempo mediante una fase de localización secuencial.

La inicialización es resuelta en esta tesis para cualquier número de cámaras, aplicando un enfoque de reconstrucción a partir del movimiento del robot. Este enfoque requiere el uso de odometría en su versión para una cámara y permite que las correspondencias entre cámaras no sean necesarias.

El algoritmo secuencial de posicionamiento hace uso del modelo tridimensional obtenido en la inicialización, para determinar la pose del robot mediante un esquema probabilístico de estimación-corrección. El algoritmo incluye técnicas robustas de eliminación de medidas erróneas en el plano imagen y una solución no iterativa y de baja complejidad para el problema del cálculo de la pose utilizando múltiples cámaras, denominado el problema mPnP (“multiple Perspective n Point”) en la literatura anglosajona.

El uso de un enfoque probabilístico permite realizar la fusión de toda la información disponible en el entorno, modelando la incertidumbre resultado de la pose del robot.

La solución presentada en esta tesis se ha validado experimentalmente en todas sus fases, utilizando para ello tanto datos simulados como un entorno real con cámaras, robots y obstáculos. El resultado es un método de localización estable y aplicable a entornos reales, en los que existen oclusiones y cambios de iluminación.

Abstract

This thesis deals with the problem of mobile robot localization using static cameras placed in the environment. The presented approach is based in the idea of “Intelligent Space”, where a distributed intelligence controls cameras and robots to serve in a certain task. The previous works that shares the same approach are focused on placing artificial landmarks on the robots. This thesis focuses on approaches that do not need previous knowledge about the robot and make use of the natural appearance of the robot.

The localization process proposed in this thesis is based on natural landmarks, which are detected in the image plane of the set of cameras and correspond to a 3D model of the robot.

The proposed localization system is divided in two steps. Firstly, a initialization step obtains the 3D model of the robot and its initial pose. Secondly, using a sequential approach, the pose of the robot is obtained at each time instant.

The initialization step is solved in this thesis for any number of cameras using a structure-from-motion approach, where odometry serves as a metric reference in the single camera case. Besides, the proposed approach avoids using natural landmark correspondences between multiple cameras, which allows to initialize the 3D model for non-overlapping views.

The sequential step proposed in this thesis uses the 3D model, obtained in the aforementioned initialization step, for retrieving robot’s pose in a estimation-correction scheme. This step includes robust techniques that remove outliers from the measurements.

In addition to the filtering approach, a non-iterative and of low complexity solution to the mPnP (multiple perspective n point) problem is proposed.

The probabilistic approach performs coherent data fusion between all information available and it allows to estimate uncertainty in the obtained robot’s pose.

The solution presented in this thesis has been experimentally assessed using synthetic generated data and experiments from a real environment with cameras, robots and obstacles. The resulting method is proved to be stable against occlusions and illumination changes, which makes it suitable to real situations.

Índice general

1. Introducción a la tesis	1
1.1. Introducción	3
1.2. Motivaciones de la Tesis	3
1.3. Objetivos planteados en la Tesis	5
1.4. Contexto de desarrollo de la Tesis	7
1.5. Estructura general del algoritmo propuesto	8
1.6. Estructura de la Tesis	9
2. Revisión de conocimientos	11
2.1. Introducción	13
2.1.1. Definición del problema	16
2.2. Seguimiento de objetos rígidos basado en marcas artificiales	18
2.2.1. Marcas artificiales puntuales	18
2.2.2. Marcas artificiales planas	20
2.3. Seguimiento de objetos rígidos mediante marcas naturales	20
2.4. Marcas naturales en visión monocular	22
2.4.1. Métodos basados en modelo previo	22
2.4.2. Métodos simultáneos de obtención del modelo	29
2.5. Marcas naturales en visión de múltiples vistas	30
2.5.1. Métodos basados en descriptores invariantes	31
2.6. Conclusiones	34
3. Estructura y modelado matemático de los elementos del sistema	37
3.1. Introducción	39
3.2. Robot móvil	39
3.2.1. Pose del robot	41
3.2.2. Modelos de movimiento del robot	42
3.2.3. Modelo geométrico de la estructura del robot	42
3.3. Modelo y distribución de cámaras	43
3.3.1. Modelo de cámara	44
3.3.2. Vector de medidas en el plano imagen	45
3.4. Desc. de la Pose y la Proyección en la Imagen como Proc. Aleatorios	46
3.4.1. Descripción de la pose del robot como un proceso aleatorio	47
3.4.2. Proceso aleatorio conjunto de pose y geometría	50
3.4.3. Proceso de medida en la imagen como proceso aleatorio	50
3.5. Conclusiones	53

4. Detección de marcas naturales y procesos de seguimiento	55
4.1. Introducción	57
4.1.1. Situaciones planteadas en la tesis	57
4.2. Detección de marcas naturales	59
4.2.1. Detectores de regiones o “blobs” en un espacio de escala	61
4.3. Mecanismo de correspondencia de marcas naturales	62
4.3.1. Correspondencia de puntos en la fase de inicialización del sistema . .	64
4.3.2. Correspondencia de marcas naturales en procesos de Estimación-Corrección	75
4.3.3. Correspondencias “Wide-Baseline”	79
4.4. Conclusiones	84
5. Inicialización de la pose y geometría	87
5.1. Introducción	89
5.2. Inic. de la pose y geom. del robot utilizando una única cámara	90
5.2.1. Odometría y medidas en el plano imagen	92
5.2.2. Enfoque de máxima verosimilitud	97
5.2.3. Influencia del modelado estadístico en la solución	101
5.2.4. Error de reconstrucción	102
5.2.5. Tiempo de cómputo	111
5.2.6. Tratamiento de los “outliers”	112
5.2.7. Inicialización de la solución previa a la optimización	115
5.2.8. Obtención de la distribución gaussiana de la solución	121
5.2.9. Configuraciones degeneradas	122
5.2.10. Evaluación experimental del método propuesto	123
5.3. Inicialización de la pose y geometría del robot mediante múltiples cámaras .	128
5.3.1. Definiciones preliminares	129
5.3.2. Función de coste	130
5.3.3. Tratamiento de los “outliers”	133
5.3.4. Tiempo de cómputo	133
5.3.5. Inicialización de la solución previa a la optimización	134
5.3.6. Obtención de la distribución gaussiana de la solución	138
5.3.7. Configuraciones degeneradas	138
5.3.8. Evaluación experimental del método propuesto	138
5.4. Conclusiones	140
6. Obtención de la pose del robot a partir de múltiples cámaras	145
6.1. Introducción	147
6.2. Prop. no iterativa de pose a partir de mult. cámaras	148
6.3. Parametrización en coordenadas baricéntricas	149
6.4. Construcción del sistema lineal	152
6.4.1. Sistema sobredeterminado o perfectamente determinado	155
6.4.2. Sistema con un único valor singular nulo	156
6.4.3. Sistema con dos valores singulares nulos	159
6.4.4. Sistema con más de dos valores singulares	160
6.5. Solución para movimientos restringidos a un plano	163

6.5.1.	Construcción del sistema lineal	163
6.5.2.	Sistema sobredeterminado o perfectamente determinado	164
6.5.3.	Cómputo de pose para puntos coplanares	165
6.5.4.	Cómputo de pose para puntos colineales	166
6.6.	Análisis de la complejidad numérica del método	166
6.7.	Efectos del ruido de la medida y el modelo en la estimación de la pose . . .	167
6.7.1.	Ruido y autovalores de la matriz $A^T A$	167
6.7.2.	Propagación del ruido de medidas y el modelo en la estimación . . .	168
6.8.	Evaluación experimental	172
6.9.	Conclusiones	175
7.	Localización secuencial de robots móviles	177
7.1.	Introducción	179
7.1.1.	Descripción de la propuesta de obtención secuencial de la pose . . .	179
7.2.	Proceso de inferencia secuencial	181
7.3.	Método “online” de cómputo de pose a partir de múltiples cámaras	183
7.3.1.	Fase de estimación	183
7.3.2.	Fase de corrección	183
7.3.3.	Complejidad algorítmica	184
7.4.	Método secuencial de cálculo de pose de baja complejidad	184
7.4.1.	Vector de medidas	185
7.4.2.	Etapas de estimación y corrección	185
7.4.3.	Complejidad algorítmica	188
7.5.	Capa robusta	188
7.5.1.	Descripción del algoritmo RANSAC	189
7.6.	Evaluación experimental del método propuesto	190
7.7.	Inclusión de nuevos puntos en el vector de estados	193
7.7.1.	Inclusión de geometría en el vector de estados utilizando una única cámara	193
7.8.	Inicialización de la pose o problema del secuestro	199
7.9.	Conclusiones	201
8.	Resultados y validación del sistema propuesto	203
8.1.	Introducción	205
8.2.	Elementos que componen el “espacio inteligente”	205
8.2.1.	Descripción de las cámaras	205
8.2.2.	Plataforma robótica	207
8.2.3.	Arquitectura distribuida de adquisición y procesado	208
8.3.	Experimento utilizando una cámara	208
8.3.1.	Experimento de inicialización	210
8.3.2.	Experimento de localización secuencial	212
8.3.3.	Experimento en una secuencia que incluye oclusiones	212
8.4.	Experimento utilizando un entorno de múltiples cámaras	215
8.5.	Conclusiones	219

9. Conclusiones	221
9.1. Introducción	223
9.2. Conclusiones y aportaciones de la tesis	223
9.3. Líneas de investigación futuras derivadas de la tesis	224

Capítulo 1

Introducción a la tesis

1.1. Introducción

En las últimas décadas, las tecnologías de la información se han establecido definitivamente en la práctica totalidad de los ámbitos de la sociedad humana. Las facilidades generadas por las redes de datos, desarrolladas gracias al tremendo avance de las comunicaciones en el último siglo, han permitido sentar las bases para un futuro nivel de interacción entre el ser humano y la tecnología. Mediante una red densa y flexible de comunicaciones, como lo es la red global existente, es exigible una futura comunicación e interacción con el usuario y el entorno igualmente rica.

Una parte amplia de la producción científica, como es la robótica, los interfaces hombre-máquina o las redes de sensores, desarrolla en la actualidad con gran velocidad el objetivo de dotar de inteligencia y poder de decisión a los sistemas que ahora sustentan las redes de información. En este marco visionario se desarrollan los llamados “Espacios Inteligentes”, herederos de la computación ubicua, en los cuales la tecnología se encuentra inmersa en los espacios ocupados por el usuario, poniendo al servicio del mismo interacción con el entorno. El “Espacio Inteligente” tiene a su disposición una red de sensores que le permiten obtener información del mundo que observa, y una red de actuadores (robots, pantallas de información, etc.) que habilitan su interacción con el usuario. La red de sensores y la inteligencia que los gobierna es por tanto imperceptible para el usuario, tomando decisiones y cooperando entre diferentes agentes autónomos para la consecución de un servicio o tarea.

El concepto de “Espacio Inteligente” aglutina los conceptos de ubicuidad e inteligencia de las redes sensoriales. La computación ubicua, propuesta por Weiser, engloba el concepto de inmersión y transparencia de la tecnología para el usuario. Existen numerosas acepciones para el mismo concepto, y por lo tanto se habla igualmente de “Ambientes Inteligentes” o “Smart Rooms” en la terminología anglosajona.

En la presente tesis se considera a un “Espacio Inteligente” como un área física dotada de sensores, los cuales son controlados e inspeccionados por un sistema de supervisión con capacidad de análisis y toma de decisiones. La inteligencia del entorno puede estar constituida por uno o varios nodos interconectados que permiten la captura y análisis de la información extraída por los sensores. Asimismo, los actuadores disponibles, entre los que destaca el uso de robots móviles, son convenientemente controlados por dicha inteligencia para acometer una tarea concreta. Las aplicaciones de los “Espacios Inteligentes” son numerosas y de emergente aplicabilidad e implantación práctica. Abarcan todos los ámbitos de la robótica de servicios, tareas de seguridad y vigilancia automática, ayudas a personas con discapacidad y por supuesto los servicios domóticos orientados al gran público.

1.2. Motivaciones de la Tesis

Una de las tareas fundamentales que los “Espacios Inteligentes” deben resolver satisfactoriamente, en especial si su interacción se realiza con robots móviles, es la correcta localización de todos los elementos que se encuentran en una escena particular. La localización implica no solo una posible medida métrica de la posición y según el caso orientación del objeto a considerar, sino también el desarrollo de métodos capaces de discernir la iden-

tividad de los diferentes elementos percibidos por los sensores.

La dificultad y posibilidades de la tarea de localización es dependiente de las tecnologías involucradas en el sistema sensorial. Existe una actividad importante de investigación la cual, se orienta a crear redes de sensores para proporcionar localización en entornos interiores y exteriores. En este punto y de manera natural los “Espacios Inteligentes” se funden con dicha actividad investigadora, compartiendo las mismas problemáticas y objetivos.

En la actualidad existen diversas tecnologías, con diferentes grados de implantación y objetivos, dedicadas a la localización en espacios interiores y exteriores. En la figura 1.1 se muestra el grado de implantación de un conjunto relevante de sistemas de localización en relación con la precisión obtenida. El posicionamiento en exteriores posee un grado de implantación y éxito notables, debido en gran parte a la evolución y generalización de las tecnologías GPS y similares. Sin embargo, en entornos interiores la situación es muy diferente. Las tecnologías con mayor grado de implantación en interiores, como puede ser el posicionamiento por RFID [Hahnel et al., 2004] o WiFi [Ocana et al., 2005], son difícilmente aplicables al posicionamiento métrico de robots, debido en parte a la precisión obtenida y su dependencia con la presencia de personas y otros objetos en el entorno.

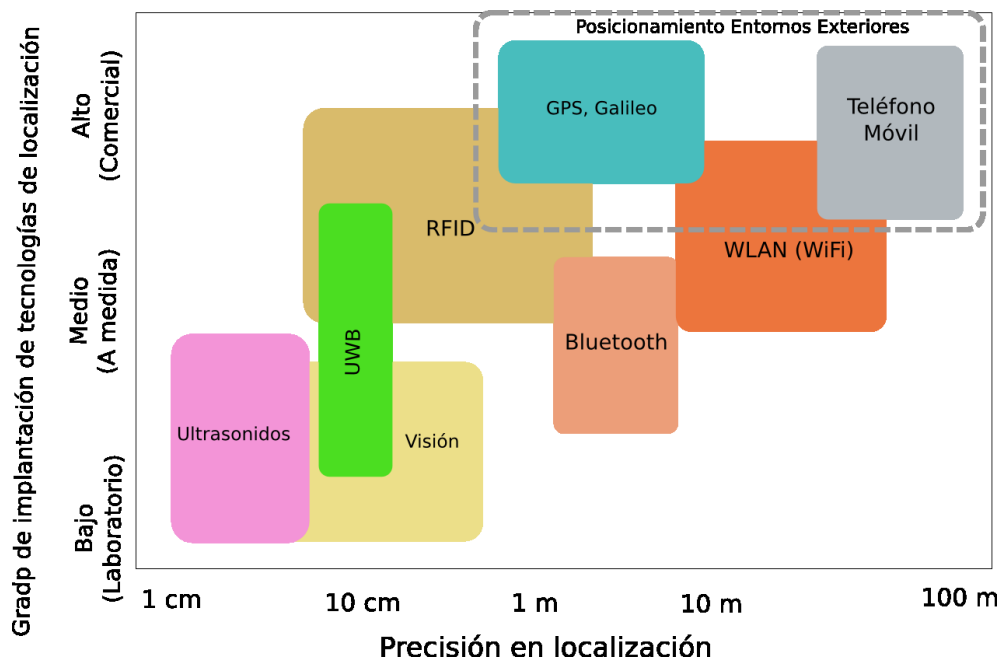


Figura 1.1: Clasificación de las técnicas de localización atendiendo a la precisión alcanzable y su grado de implantación. La altura de las cajas de cada tecnología intenta representar la evolución que se estima tendrá en los próximos años [Hazas et al., 2004].

A su vez, existen otras tecnologías con un grado de implantación menor, como puede ser el “Ultra Wide Band (UWB)” [Ahn and Yu, 2006], los ultrasonidos [Villadangos et al., 2005] o la visión [Fernandez et al., 2007], que presentan en la actualidad resultados prometedores para el posicionamiento preciso (del orden de centímetros).

Entre el conjunto de tecnologías comentadas cabe destacar el uso de visión artificial por su potencial en varios aspectos representativos:

- Proporciona una cantidad de información muy superior a las tecnologías como UWB, acústica o infrarrojos.
- Su uso no sólo está limitado al posicionamiento sino también al reconocimiento de objetos y gestos de los usuarios.
- El uso de cámaras permite dotar al “Espacio Inteligente” de un sistema sensorial enteramente transparente al usuario.
- La precisión alcanzada por la visión es del orden de centímetros, lo que permite su uso para guiado de robots móviles.

Como inconvenientes del uso de cámaras, lo cual ha supuesto un freno en su implantación directa fuera del laboratorio, se encuentran los siguientes:

- El uso de cámaras exige el tratamiento de una cantidad considerable de información ambigua y compleja.
- Las necesidades de hardware aumentan y el tiempo real requiere de un uso óptimo de la información proporcionada.
- El coste de un sistema basado en visión artificial, el cual a pesar de una constante disminución en los últimos años, se encuentra aún por encima de soluciones comerciales basadas en otras tecnologías.
- La cantidad de información suministrada por cámaras genera en muchos casos problemas de protección de privacidad, sobre todo en su aplicación en espacios públicos.

Los inconvenientes del uso de cámaras como sensor de posicionamiento se deben por tanto en su mayoría a limitaciones en la tecnología existente y no a las posibilidades potenciales que posee. La visión por computador es una disciplina aún emergente en ciertos aspectos, que sin lugar a dudas terminará imponiéndose en numerosas aplicaciones, hoy ocupadas por otras tecnologías.

1.3. Objetivos planteados en la Tesis

La presente tesis se centra en el uso exclusivo de cámaras para la obtención de la localización de los robots móviles que son controlados por el “Espacio Inteligente”. Éste es dotado, por tanto, de un conjunto de cámaras dispuestas para la observación de un espacio acotado. Cada cámara se conecta a una red de comunicaciones, que permite la adquisición de imágenes sincronizadas por un conjunto de nodos de procesado. Además de las cámaras, el entorno controla robots móviles mediante un enlace inalámbrico igualmente interconectado con todos los nodos inteligentes. En la figura 1.2, se muestra un gráfico de la estructura del “Espacio Inteligente”.

Las condiciones desde las que parte la tesis son las siguientes:

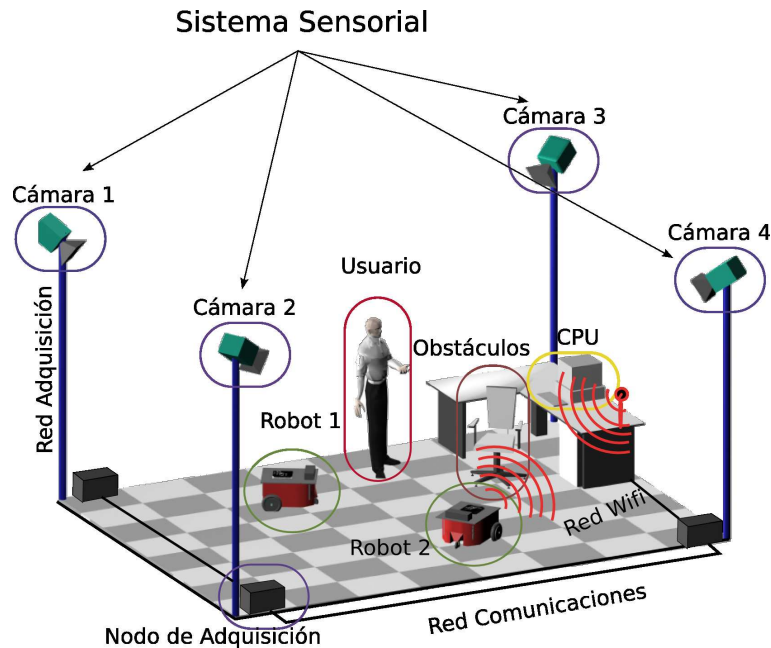


Figura 1.2: Estructura del espacio inteligente

- Cada robot, del que se requiere una localización precisa, es controlado remotamente por el “Espacio Inteligente”. Dicha controlabilidad será una de las características fundamentales que lo diferenciará del resto de objetos móviles presentes en la escena.
- No está disponible un modelo de la estructura tridimensional del robot y por lo tanto no es necesario que esté equipado con ningún tipo de marca artificial o sistema de balizamiento. Se desea un sistema totalmente no invasivo con los elementos que localiza.
- Se dispone de un sistema de estimación de movimiento odométrico, embarcado en el robot, el cual es enviado al “Espacio Inteligente” mediante un enlace inalámbrico. El sistema odométrico será indispensable en ciertos aspectos de la solución planteada, y en cualquier caso ayudará a mejorar el comportamiento y estabilidad de la propuesta.
- El sistema sensorial se compone de una o varias cámaras. La solución desarrollada en esta tesis cuenta con una disposición de los sensores alejados, de modo que sea posible cubrir el máximo posible de espacio con el mínimo número de ellos.

Los principales problemas a los que se hace frente en esta tesis se resumen a continuación:

- Diseñar un mecanismo para detectar características robustas de la proyección del robot en cada una de las imágenes disponibles. Éstas deben permitir realizar una localización adecuada gracias a su seguimiento y correspondencia a lo largo del tiempo.
- Fusión de la información proveniente de todos los sensores presentes en el entorno.

- Presencia de oclusiones, cambios de iluminación y sombras en la obtención de la información de las distintas cámaras.
- Diseño de un algoritmo capaz de realizar la localización simultánea de múltiples robots.
- Cada paso del proceso deberá ser resuelto para distintas situaciones de cobertura del espacio, incluyendo aquellas en las que tan solo se cuenta con información proveniente de una sola cámara.
- Definición y solución a los problemas derivados de la inicialización.

1.4. Contexto de desarrollo de la Tesis

Esta tesis se enmarca dentro de las actividades de investigación que se realizan en el grupo de investigación GEINTRA (Grupo de Ingeniería aplicada a los Espacios Inteligentes y el Transporte), perteneciente al Departamento de Electrónica de la Universidad de Alcalá. Dicho grupo posee una línea de investigación en “Espacios Inteligentes” aplicados al guiado y posicionamiento de robot móviles.

En concreto, la presente tesis se ha desarrollado dentro de los siguientes proyectos:

- *“Integración de Redes de Sensores Acústicos, de Visión y RFID para la localización en Ambientes Inteligentes” (RESELAI)* Proyecto financiado por el Ministerio de Ciencia y Tecnología de España (Ref. TIN-2006-14896-C0201).

El objetivo del proyecto RESELAI es el estudio, desarrollo e integración de redes de sensores a partir de tres tecnologías básicas: acústica, visión y RFID, con el fin de potenciar entornos inteligentes, a los que aportar funciones de detección de presencia, localización física e interacción con sus usuarios. En cada una de estas áreas el objetivo es el desarrollo de los sistemas “software” y “hardware” necesarios, con la finalidad de construir un prototipo funcional. Éste debe permitir demostrar las facetas de detección, localización e integración mejorada con los usuarios (personas, sillas de ruedas automáticas, robots móviles, etc.).

El objetivo del proyecto RESELAI consiste no sólo en desarrollar las tres tecnologías aplicadas a “Espacios Inteligentes” por separado, sino presentar a su vez un esquema de sinergia eficiente entre ellas.

- *“Espacios Inteligentes para la Identificación de Robots Móviles utilizando Visión Artificial” (VISO)* Proyecto financiado conjuntamente por la Universidad de Alcalá y la Comunidad de Madrid (Ref. CCG06-UAH/DPI0745).

El proyecto VISO tiene como objetivo la investigación de técnicas no invasivas, basadas exclusivamente en visión, de posicionamiento de robots móviles en “Espacios Inteligentes”. En este proyecto se hace hincapié en técnicas de localización basadas en marcas naturales de la apariencia de los robots y técnicas de segmentación basadas en movimiento.

1.5. Estructura general del algoritmo propuesto

En la presente tesis se plantea la localización de un robot mediante la identificación de puntos de su estructura natural en el conjunto de cámaras. Se construirá un modelo tridimensional de dicha estructura, con el objetivo de relacionar métricamente la pose del robot con la proyección en la imagen o imágenes de dicho modelo. Entre los objetivos de la tesis, se encuentra la no posibilidad de no balizar el robot y el desconocimiento de la estructura del mismo. Por lo tanto, se plantea la obtención no supervisada del modelo tridimensional, necesario para calcular la pose, mediante el conjunto de cámaras. El mecanismo de obtención de información tridimensional mediante cámaras se suele denominar como “reconstrucción” y por tanto, el objetivo de la tesis es la reconstrucción y posicionamiento de robots móviles mediante cámaras. En la figura 1.3 se resumen los diferentes procesos involucrados en la solución propuesta que calcula la pose de robots móviles en “Espacios Inteligentes”.

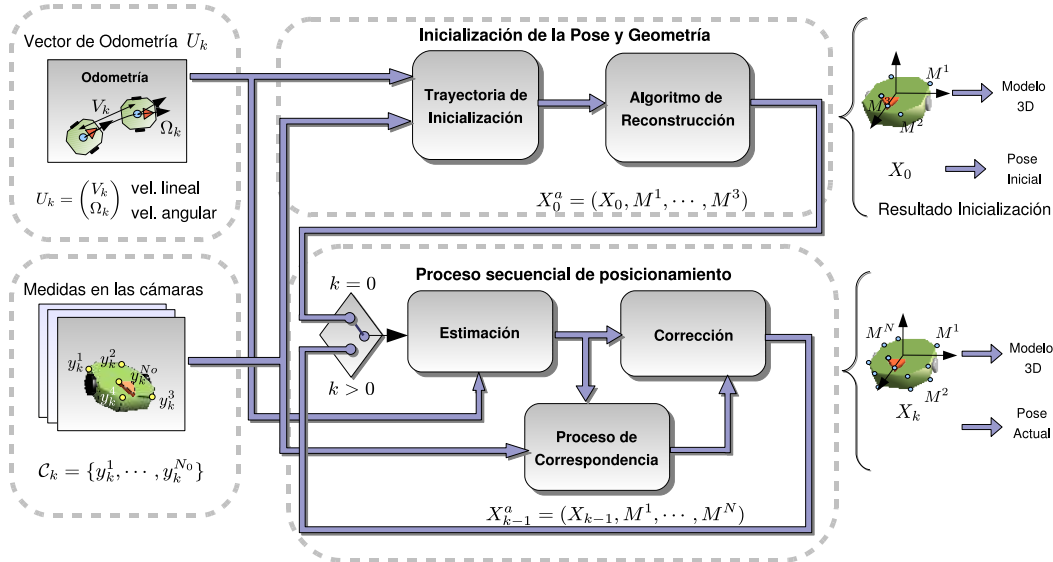


Figura 1.3: Esquema de los procesos involucrados en el cálculo de la pose del robot móvil.

La reconstrucción de un modelo tridimensional del robot móvil se llevará a cabo al mismo tiempo que se localiza el robot, haciendo uso para ello de la información proveniente de las cámaras y las estimaciones de movimiento proporcionadas por sensores de odometría instalados en cada robot, los cuales son remotamente supervisados por el “Espacio Inteligente”. El algoritmo desarrollado en esta tesis posee una división clara en dos tipos de procesos: En primer lugar se proponen aquellos procedimientos destinados a proporcionar información en los estados de arranque del sistema (“Inicialización de la Pose y Geometría”). Resuelta la inicialización, y por tanto conocida la pose anterior y una porción de la estructura del robot, se proponen aquellos métodos destinados a obtener la pose y reconstrucción de manera secuencial (“Posicionamiento y Reconstrucción Secuencial”).

La propuesta presentada en esta tesis para el posicionamiento de robots móviles consta, por tanto, de los siguientes bloques principales (ver figura1.3):

- **Inicialización de la Pose y Geometría:** Inicialmente, no es conocida ni la pose del robot ni información tridimensional del mismo. En este paso, la información del conjunto de cámaras y la estimación de movimiento proporcionada por la odometría del robot, son utilizadas para obtener la pose y reconstrucción del mismo. La dificultad de esta fase reside en la necesidad de resolver problemas que consumen un tiempo de cómputo elevado o incluso requieren que el robot realice una trayectoria de inicialización.
- **Posicionamiento y Reconstrucción Secuencial:** Una vez resuelto el problema de la inicialización, el problema secuencial consiste en la obtención de la pose y nueva información tridimensional del robot, dado que se conoce la pose y estructura en el instante anterior. El enfoque secuencial requiere menor tiempo de cómputo que el problema de la inicialización, lo que permite su implementación en tiempo real.

Ambos procesos, inicialización y secuencial, se nutren de la misma información proveniente de las cámaras dispuestas en el “Espacio Inteligente”:

- **Proceso de medida en el plano imagen:** Las medidas en el plano imagen consisten en puntos asociados con la proyección de la estructura reconstruida del robot. El proceso de medida tiene dos objetivos fundamentales:
 - Detectar los puntos de la imagen que son susceptibles de ser candidatos a formar parte de la estructura del robot.
 - Establecer una correspondencia de un mismo punto en sucesivas imágenes.
- **Estimación de movimiento mediante odometría:** La odometría presente en el robot permite establecer una estimación, precisa en trayectorias cortas, del movimiento realizado por el robot, el cual es ordenado por el propio “Espacio Inteligente”.

1.6. Estructura de la Tesis

La organización de la memoria de esta tesis atiende al desarrollo y justificación de todos los conceptos teóricos presentados, así como la validación experimental efectuada sobre cada una de los temas que se han propuesto para su desarrollo. La memoria de la presente tesis está estructurada en nueve capítulos.

En el **capítulo 1** se realiza una breve introducción a los “Espacios Inteligentes” y las redes de sensores. Se realiza una justificación de las motivaciones que dan lugar a la presente tesis, así como un breve esbozo de la solución planteada para la localización de robots móviles mediante cámaras.

En el **capítulo 2** se presenta un estudio del estado del arte actual, tanto en lo que a

“Espacios Inteligentes” se refiere como a las últimas técnicas disponibles en visión artificial para la detección y posicionamiento de objetos.

En el **capítulo 3** se describen los modelos matemáticos de todos los elementos que componen el “Espacio Inteligente” así como la notación establecida para el resto del documento. Se plantea a su vez la descripción de todos los procesos dinámicos y de medida como procesos aleatorios con funciones de densidad de probabilidad gaussianas. El objetivo es la introducción de las técnicas probabilísticas que serán utilizadas en apartados posteriores.

En el **capítulo 4** se describen los procesos de inicialización de pose y estructura del robot para una o varias cámaras. Se describen en detalle los algoritmos utilizados, su impacto en el tiempo de cómputo, precisión y robustez.

En el **capítulo 5** es presentado el método de medida en el plano imagen mediante métodos de detección y seguimiento de marcas naturales. En este capítulo se presta especial atención sistemas de emparejamiento de marcas naturales basado en descriptores invariantes a deformaciones proyectivas.

En el **capítulo 6** se describe una técnica de cómputo de pose de un objeto a partir de múltiples cámaras, dado un modelo tridimensional del mismo. El método planteado será de baja complejidad para un número elevado de puntos vistos por las cámaras. Dicha técnica se utilizará como algoritmo básico en su utilización en el capítulo séptimo.

En el **capítulo 7** se presenta el algoritmo secuencial de obtención de la pose y geometría del robot, partiendo de una solución inicial conocida. Este capítulo aborda aspectos del algoritmo, como el tiempo de cómputo o su inmunidad a medidas erróneas.

En el **capítulo 8** se muestran los resultados experimentales que permiten validar la propuesta de localización presentada en esta tesis.

En el **capítulo 9** se presentan las conclusiones finales junto a un resumen de las aportaciones realizadas en esta tesis. Así mismo, se indican futuras líneas de investigación, relacionadas con el trabajo desarrollado.

Capítulo 2

Revisión de conocimientos

2.1. Introducción

El concepto de un espacio dotado de inteligencia ha sido propuesto por diversos autores y ha ido evolucionando separándose en distintas implementaciones, incluyendo aquellas aplicaciones destinadas a la localización y navegación de robots; relacionándose directamente con el tema propuesto en esta tesis.

El enfoque de “Espacio Inteligente” proviene de áreas de investigación relacionadas con el interfaz hombre-maquina (HMI). El objetivo es definir espacios donde exista una interacción no intrusiva entre usuarios y el propio entorno. La interacción tiene la intención de ayudar al usuario en algún tipo de tarea. En un entorno, el usuario humano recibe servicios del mismo mediante una comunicación que mantiene con él empleando para ello un lenguaje natural entendido por ambos.

El inicio de los espacios inteligentes y la “Computación Ubicua” fueron conducidos por Weiser [Weiser, 1999] [Weiser, 1993b]. Este autor realiza una clasificación en función de cuatro elementos básicos: ubicuidad, conocimiento, inteligencia e interacción natural.

- *Ubicuidad*: Múltiples sistemas embebidos con total interconexión entre ellos
- *Conocimiento*: Habilidad del sistema para localizar y reconocer lo que acontece en el entorno y su comportamiento.
- *Inteligencia*: Capacidad de adaptarse al mundo que percibe.
- *Interacción Natural*: Capacidad de comunicación entre el entorno y los usuarios

Uno de los primeros sistemas ubicuos implementados con esta filosofía se propuso en el proyecto Xerox PARC de Computación Ubicua (UbiComp) [Weiser, 1993a], desarrollado a finales de los 80. La red de sensores desplegada no estaba basada en información visual debido al coste prohibitivo para la época. Hoy en día varios grupos de investigación desarrollan y expanden el concepto de espacio inteligente y la computación ubicua. Algunos de los mas notorios se indican a continuación:

- *Intelligent Room* [Coen, 1998]. Desarrollado por el grupo de investigación del *Artificial Intelligence Laboratory* en el MIT (Masachusetts Institute of Technology). Es uno de los proyectos mas evolucionados en la actualidad, bajo el nombre del proyecto “Oxygen” [Look et al., 2005] [Look and Shrobe, 2007]. Se trata de una habitación que posee cámaras, micrófonos y otros sensores que realizan una actividad de interpretación con el objetivo de averiguar las intenciones de los usuarios. Se realiza interacción mediante voz, gestos y contexto (ver figura 2.1).
- *Smart Room* [Pentland, 1996]. Desarrollado en el *Media Lab* del MIT bajo el grupo de investigación VisMod (“Vision and Modelling”). Mediante la utilización de cámaras, micrófonos y sensores se pretende analizar el comportamiento humano dentro del entorno. Se localiza al usuario, identificándolo mediante su voz y apariencia y se realiza un reconocimiento de gestos. La línea de investigación actual se centra en la utilización de redes de sensores para su aplicación a productividad en grandes empresas [Ara et al., 2008].

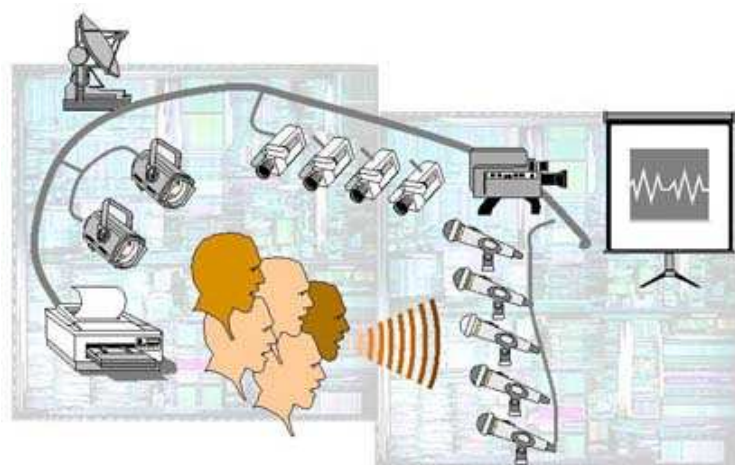


Figura 2.1: Diagrama del proyecto oxygen realizado en el MIT.

- *Easy Living* [Shafer et al., 1998]. Se trata de un proyecto de investigación de la empresa Microsoft con el objetivo de desarrollar “entornos activos”, que ayuden a los usuarios en tareas cotidianas. Al igual que otros proyectos comentados, se intenta establecer una comunicación entre el entorno y el usuario mediante lenguaje natural. Se hace uso de visión artificial para realizar identificación de personas e interpretación de comportamientos dentro del espacio inteligente. El proyecto no posee actividad investigadora relevante desde el año 2000.

Los proyectos mencionados no incluyen robots controlados por el entorno. El uso de agentes robóticos controlados por el espacio inteligente es estudiado por un número todavía reducido de grupos de investigación.

- T.Sogo [T.Sogo et al., 1999] propone un sistema equipado con 16 cámaras fijas llamadas “Agentes de Visión” (VA), las cuales aportan la información necesaria para localizar a un grupo de robots a través de un espacio lleno de obstáculos. Los Agentes de Visión consisten en cámaras omnidireccionales no calibradas. La localización se realiza en dos pasos. En primer lugar un operador humano debe mostrar al espacio inteligente el camino que deberán seguir los robots. Una vez que el sistema ha aprendido en coordenadas de imagen el camino señalado, puede actuar sobre los robots con el objetivo de minimizar el camino que recorren en la imagen con respecto al que realizan durante el periodo supervisado.

La técnica es similar a la utilizada en aplicaciones de servo óptico, por lo que resulta muy difícil estimar la precisión que se obtiene realmente con el sistema. Dependerá en gran medida de la técnica de comparación en el plano imagen y la posición relativa de las cámaras y los robots.

- Un trabajo más evolucionado fue propuesto en la universidad de Tokyo por Lee y Hashimoto [Hashimoto et al., 2003] [Lee et al., 2005]. En este caso se dispone de un Espacio Inteligente completo en el que se mueven robots y usuarios. Cada cámara, unida a un sistema de procesamiento es tratado como un dispositivo inteligente

conectado a una red, denominado DIND (Distributed Intelligent Network Device). Cada uno de los DIND posee capacidad de procesamiento por sí solo, por lo que el espacio inteligente posee flexibilidad a la hora de incluir nuevos nodos inteligentes. Puesto que se cuenta con dispositivos calibrados, la localización se realiza en espacio métrico. Los robots están dotados de balizamiento pasivo mediante un conjunto de bandas de colores fácilmente detectables por cada DIND. En este espacio de trabajo (Ispace), se realizan experimentos de interacción entre humanos y robots y navegación con detección de obstáculos.

- Otra propuesta es el proyecto MEPHISTO (“Modular and Extensible Path Planning System using Observation”) [Steinhaus et al., 1999] [Steinhaus et al., 2007] del Instituto para el Diseño de Computadores y Tolerancia a Fallos en Alemania. En este proyecto, el concepto de inteligencia distribuida se alcanza gracias a las denominadas Unidades de Procesamiento de Área Local (LAPU, “Local Area Processing Unit”), similares a los DIND de Lee y Hashimoto. Se define una unidad específica para el control del robot (RCU, “Robot Control Unit”) que conecta y envía instrucciones a los diferentes robots. La localización se realiza sin marcas artificiales, usando una descripción poligonal de cada robot en coordenadas de la imagen, la cual es convertida a una posición tridimensional con un enfoque multicámara.
- En el Departamento de Electrónica de la Universidad de Alcalá existe un grupo de “Espacios Inteligentes” (grupo GEINTRA, “Grupo de Espacios Inteligentes y Transporte”) [Villadangos et al., 2005] [Pizarro et al., 2005] [Fernandez et al., 2007] en el que, mediante un conjunto de sensores, se pretende realizar posicionamiento de robots móviles. Las alternativas incluyen visión artificial, ultrasonidos, infrarrojos y voz. En dicho grupo de investigación se enmarca la presente tesis.

Los proyectos anteriormente comentados hacen hincapié en el diseño de sistemas distribuidos y flexibles, donde la inclusión de un nuevo sensor al entorno se realiza de forma dinámica sin afectar al funcionamiento del sistema. Se han desarrollado a su vez trabajos, que si bien se diferencian en el planteamiento genérico, las técnicas usadas y el objetivo se relacionan directamente con la tesis propuesta. Entre dichos trabajos, se pueden citar los siguientes:

- Destaca entre otros el trabajo realizado por Hoover y Olsen [Hoover and Olsen, 2000], en el que utilizando un conjunto de cámaras con coberturas solapadas, son capaces de detectar obstáculos estáticos y dinámicos. Mediante un mapa de ocupación [Hoover and Olsen, 1999] es posible guiar un robot dentro del espacio de cobertura conjunta.

La técnica del mapa de ocupación requiere una fase previa de calibración, en la cual el espacio se encuentra libre de obstáculos. Se obtienen imágenes de referencia que serán utilizadas para la localización de obstáculos en posteriores capturas.

Mediante tablas de búsqueda se obtiene una correspondencia directa entre puntos de pantalla y las coordenadas tridimensionales de puntos pertenecientes al plano que representa el suelo del espacio. Realizando un fundido de la información de las distintas cámaras, se obtiene un mapa de aquellos puntos del suelo que tienen alta probabilidad de estar ocupados.

Uno de los principales inconvenientes de este enfoque, es la detección de objetos con bajo contraste con respecto al color del suelo en la imagen de referencia. El sistema desperdicia mucha información tridimensional mediante el mapa de ocupación que podría ser utilizada para obtener información tridimensional del entorno y de los objetos que se mueven en él.

- Un proyecto similar al anterior es el presentado en [Kruse and Wahl, 1998], denominado MONAMOVE(“Monitoring and Navigation for Mobile Vehicles”) y orientado a la navegación de un vehículo en un entorno industrial. En este trabajo se propone fundir la información de localización, obtenida del conjunto de cámaras distribuidas por el entorno, con sensores de ultrasonidos e infrarrojos ubicados a bordo del robot. Al igual que el trabajo de Hoover y Olsen, se utilizan tablas de búsqueda para crear un mapa de ocupación del entorno.

Los obstáculos móviles se detectan mediante análisis diferencial en sucesivas imágenes y los estáticos son comparados con dos imágenes de referencia, obtenidas en un proceso previo de calibración. Dichas imágenes de referencia se toman bajo dos condiciones de iluminación diferentes, y se realiza una comparación ponderada con la imagen actual para obtener regiones correspondientes a objetos que existen en el entorno.

La incorporación de robots móviles al concepto de “Espacio Inteligente” es una disciplina aún emergente, en la que no se han explotado todavía las posibilidades que ofrece. Las técnicas de visión artificial aplicadas hasta la fecha, en general, requieren de balizamiento y no aprovechan toda la información de la que se dispone desde el conjunto de cámaras.

2.1.1. Definición del problema

El entorno planteado en esta tesis, a expensas de los trabajos identificados con “Espacios Inteligentes”, define un problema de tracking de objetos rígidos mediante múltiples vistas. Como se demostrará en esta sección, las condiciones planteadas sitúan a la posible solución, objetivo en esta tesis, en un punto equidistante entre varias subdisciplinas de los procesos de tracking 3D mediante cámaras.

La primera característica notoria, que restringe la solución al problema planteado, es la disposición relativa de cada una de las cámaras que existen en el entorno. En general se supone una estructura de “Espacio Inteligente” en el que las cámaras se encuentran separadas una distancia de varios órdenes de magnitud de la distancia focal de cada una de ellas. De hecho, la distancia relativa entre cámaras puede ser equivalente a la distancia a la que los objetos de interés se encuentran dispuestos con respecto a las cámaras. Este esquema, conocido como “wide-baseline”, posee ventajas notables en la precisión obtenida en procesos de triangulación y en la eficacia en el número de cámaras necesarias para cubrir un espacio acotado. Sin embargo, dichas ventajas están condicionadas a la existencia de una solución estable al problema de correspondencia entre imágenes, la cual no es inmediata en todos los casos.

Todo proceso de seguimiento o “tracking” de objetos en una escena implica la solución de los siguientes problemas:

- Extracción de información en cada una de las imágenes disponibles.
- Correspondencia de la información entre diferentes cámaras y/o diferentes instantes de tiempo.
- Obtención de la pose del objeto mediante recuperación de geometría.

La configuración “wide-baseline” dificulta en gran medida el proceso de correspondencia de puntos. La correlación existente entre imágenes tomadas por diferentes cámaras es baja debido a la notable deformación proyectiva resultado de la separación entre centros de proyección. Debido a dicha carencia de similitud entre imágenes, se debe recurrir a métodos que permitan obtener correspondencia ante distorsión proyectiva. Las soluciones existentes en la literatura se pueden dividir en dos grandes tendencias.

- Procesos de tracking utilizando marcas artificiales:

En esta alternativa se enmarcan la mayoría de trabajos desarrollados, hasta la fecha, con posicionamiento de objetos en espacios inteligentes [Fernandez et al., 2007]. El uso de marcas artificiales aporta al sistema varios grados de información según su naturaleza y diseño. Los procesos de correspondencia en estos casos se nutren directamente de la información aportada por dichas marcas, reduciendo en gran medida la complejidad algorítmica de las soluciones. Asimismo, se hace posible el diseño de marcas artificiales que ofrezcan una imagen reconocible e unívoca en el proceso de extracción de información.

- Procesos de tracking utilizando marcas naturales:

Los procesos basados en el uso de marcas naturales aprovechan la información intrínseca que todo objeto rígido en movimiento expone ante una transformación proyectiva. Su uso implica trasladar la dificultad al algoritmo, que ahora deberá discriminar no sólo la correspondencia de puntos en múltiples imágenes, sino elegir aquellos que muestran un buen comportamiento en el proceso de tracking. Las soluciones aportadas en este enfoque son innumerables y de rápido desarrollo.

El proceso de detección de objetos tridimensionales mediante múltiples cámaras se encuentra por tanto condicionado a la difícil tarea de intentar relacionar información en el plano imagen de cada cámara con el resto, a fin de recuperar información tridimensional.

Aunque enmarcadas en la anterior división, se deben destacar las numerosas técnicas especializadas en realizar el seguimiento de objetos mediante una única cámara. Dado que la recuperación de geometría mediante una sola cámara sufre de ambigüedad proyectiva, el sistema debe hacer uso de fuentes de información alternativas. Estas técnicas pueden proporcionar en un momento puntual la información necesaria para resolver la ambigüedad y referenciar los resultados en espacio métrico. Como se ha comentado, la topología del espacio inteligente planteado presentan los centros de proyección de cada cámara en una configuración “wide-baseline”. La restricción que impone dicha configuración no permite aprovechar mucha de la información presente en el plano imagen, descartando desde el principio aquellos puntos no visibles por al menos dos cámaras.

De esta manera, cobran un interés alto aquellos métodos que mediante el uso de una sola cámara son capaces de aportar información aprovechando el movimiento del objeto del que se realiza el seguimiento. El objetivo que se plantea en esta tesis es utilizar este tipo de métodos en una configuración “wide-baseline” de múltiples cámaras. De este modo cada cámara aprovechará al máximo la información altamente correlada que observa, derivada del movimiento del objeto.

La información adicional que requieren dichos métodos puede ser aportada por el conjunto de cámaras. De este modo, la configuración planteada puede ser aprovechada en su totalidad. En esta tesis se proponen métodos de fusión de información que permiten acceder a una reconstrucción métrica y permiten aprovechar la información aportada por cada sensor de una manera más eficiente.

En las siguientes secciones se presentará una revisión de las técnicas de “tracking” comentadas. Se hará especial hincapié en aquellas que permiten realizar el tracking mediante marcas naturales y se revisarán los métodos basados en una única cámara.

2.2. Seguimiento de objetos rígidos basado en marcas artificiales

La utilización de marcas artificiales es una técnica clásica en visión artificial. Su uso en calibración y fotogrametría tiene como resultado una extensa bibliografía en la que se proponen diferentes métodos de balizamiento. De entre los objetivos que se persiguen mediante la utilización de marcas artificiales, se pueden destacar los siguientes:

- Cada marca artificial debe ser unívocamente determinada por su proyección en el plano imagen de la cámara.
- El proceso de extracción de la imagen de una marca artificial debe ser sencillo y robusto.
- El diseño de una marca artificial puede garantizar un proceso de detección que sea capaz de obtener su localización con precisiones mayores que la unidad de píxel.

En general se asume que la posición tridimensional o la distancia relativa entre las marcas artificiales utilizadas es conocida. El algoritmo usado en el proceso de “tracking” variará según el grado de conocimiento que se aporta al sistema.

El uso de marcas artificiales no implica una restricción a los métodos de reconstrucción existentes. Su misión principal es asegurar un proceso de correspondencia y detección en el plano imagen robusto.

2.2.1. Marcas artificiales puntuales

Uno de los métodos clásicos de “tracking” que hace uso de marcas artificiales consiste en la utilización de regiones circulares. En éstas, su apariencia es relativamente invariante ante

deformaciones proyectivas y su centroide proporciona una posición estable que puede ser obtenida con precisión subpíxelica. Los procesos de identificación de cada marca se realiza mediante el agrupamiento en composiciones o patrones geométricos.

Muchas de las soluciones reales proponen métodos que tratan de producir una imagen de alto contraste fácilmente identificable. De esta forma existen soluciones que hacen uso de patrones esféricos o circulares a partir de materiales retro-reflectantes o luminosos. Dichos métodos permiten discriminar mediante sencillos algoritmos de umbralización o incluso filtros ópticos en el rango visible, aquellos puntos que representan una marca del resto de elementos de la escena.

En entornos controlados y asegurando un tamaño mínimo en detección, dichos métodos basados en marcas artificiales permiten obtener precisiones cercanas a 0.01 píxeles, indispensables para ciertas tareas que se centran en análisis fotométrico. Empresas como Geodetic services, Advanced Real-Time Tracking, Metronor, ViconPeak y AICON 3D, han desarrollado productos comerciales de gran precisión.

Otras soluciones que no requieren un grado muy elevado de precisión subpíxelica, se han propuesto activamente en la comunidad científica. Los más básicos incluyen el clásico Circulo Concéntrico de Contraste (CCC) que consiste en un círculo contrastado en la imagen y cuyo centroide es calculado tras un sencillo proceso de detección y filtrado. En [Hoff et al., 1996] se propone un patrón formado por cinco círculos concéntricos dispuestos en un plano donde tres de ellos son colineales. La pose del plano se determina de manera unívoca mediante la búsqueda de los cinco puntos y la identificación de los tres colineales.

[State et al., 1996] hace uso de marcas artificiales codificadas mediante color para una sencilla discriminación. Cada marca consiste en un círculo interno rodeado de un anillo. Se utilizan cuatro colores para la codificación de los puntos y anillos. De esta manera permite el uso de hasta doce marcas diferentes. Con el objetivo de aumentar la robustez del algoritmo, el proceso de detección comprueba que los centroides de los círculos exteriores e interiores coinciden en cierto grado. Si no es así se elimina la marca del proceso de obtención de pose.

En los trabajos anteriormente comentados el rango de seguimiento de un objeto depende de la capacidad de detectar con suficiente precisión el conjunto de marcas artificiales diseñadas. En [Cho et al., 1998] se propone un método multiresolución, mediante el cual se disponen marcas artificiales de diferentes tamaños en el objeto a detectar. Se trata de círculos concéntricos en diferentes escalas que permiten ajustar la precisión en un rango mayor de detección.

Las propuestas estudiadas están basadas en procesos de umbralizado para la detección del objeto en el plano imagen. Sin embargo, aquellos entornos que sufren de cambios de iluminación de suficiente intensidad, dificultan en gran medida los procesos de detección. El umbral estático utilizado puede producir la pérdida parcial o total de muchas de las marcas. En la literatura existente se proponen varias soluciones. En [Cho et al., 1998] se estudia también un proceso de reglas heurísticas que agrupa colores similares que son susceptibles de pertenecer a la misma marca artificial. El trabajo propuesto en [Naimark and Foxlin, 2002] propone un método de umbralizado del gradiente del logaritmo de la imagen. Se demuestra que este tipo de umbralizado es más inmune a cambios no uniformes de iluminación.

Con el objetivo de expandir las posibilidades y el número posible de marcas identificables en la imagen, se proponen métodos que añaden información codificada dentro de un esquema de marca artificial. En [Briggs et al., 2000] se proponen códigos identificativos como parte de la estructura de una marca artificial.

2.2.2. Marcas artificiales planas

Los métodos de extracción de correspondencias comentados, se basan en alcanzar precisión en la detección del centroide de un solo punto mediante técnicas que analizan la proyección de la figura que representa cada marca. La precisión obtenida en cada punto es independiente de la relación existente entre dichos puntos y solo es sensible a la distancia relativa entre el centro de proyección y cada una de las marcas.

En este apartado se contempla brevemente la posibilidad del uso de un enfoque de marca artificial que no se basa en una correspondencia puntual; se compone de un plano del que se puede calcular directamente su pose.

En [Koller et al., 1997] se propone un patrón ajedrezado como marca artificial. Mediante la extracción de segmentos de recta y la intersección entre ellos se obtiene un conjunto de correspondencias. Otros ejemplos de marcas basadas en planos se pueden encontrar en [Rekimoto, 1998] [Kato and Billingham, 1999] [Kato et al., 2000].

Si la marca artificial se compone de un plano y los parámetros intrínsecos de la cámara son conocidos, la pose se obtiene directamente a partir de la relación bidireccional (matriz de homografía) existente entre puntos del plano y el plano imagen.

Este tipo de detección destaca por su velocidad y robustez, obteniendo precisiones suficientes para tareas destinadas a proporcionar entornos para realidad aumentada y calibración.

2.3. Seguimiento de objetos rígidos mediante marcas naturales

El uso de marcas artificiales requiere aportar un grado de intervención “a priori” en el objeto del que se realiza el seguimiento. Comprende procesos de calibración previos y no hace uso de la información que de manera natural aparece en la imagen de un objeto en movimiento. El reto en los sistemas de visión artificial en la actualidad es realizar el seguimiento de objetos mediante información visual debida a la propia geometría y movimiento de la escena.

En general, este tipo de enfoque repercute en el tiempo de proceso. Sin embargo el diseño de métodos que extraigan un máximo de información posible permite que la disciplina de visión artificial avance hacia la verdadera interpretación de la escena que da lugar a una imagen.

El “Espacio Inteligente”, dotado con cámaras, requiere precisamente de dichos procesos de interpretación. No sólo con el objetivo de localizar robots que es capaz de controlar, sino como medida de detección de lo que acontece en el mundo que observa. El uso de marcas

artificiales reduce el conjunto de cámaras a un sistema fotométrico con reducidas capacidades de obtener más información que aquella suministrada por el balizamiento diseñado. Por esta razón, está justificado el uso y aportación de técnicas basadas en información presente en la escena sin intervención para la detección de objetos.

Los métodos basados en marcas naturales comprenden una inmensa bibliografía que nace en los inicios de la visión artificial y llega hasta nuestros días. Se puede considerar que su búsqueda se origina con la propia disciplina de visión y condensan grandes esfuerzos y aportaciones.

Las soluciones propuestas varían dependiendo de la aplicación concreta, las condiciones iniciales y las restricciones de funcionamiento. La organización de los métodos basados en marcas naturales debe contemplar una serie de factores importantes para su clasificación, entre los que se destacan:

- Grado de conocimiento previo que se aporta al sistema: En general, el grado de conocimiento previo está correlado con la cantidad de información geométrica que el propio sistema visual es capaz de obtener.

En el caso de sistemas que cuentan con una sola cámara, la aportación de conocimiento “a priori” es indispensable, ya sea en forma de modelo tridimensional previo o referencias métricas en la imagen.

Sin embargo, si el sistema es capaz de aportar información tridimensional directamente, el grado de conocimiento previo no se convierte en una tarea crucial para el funcionamiento del sistema. En este caso, el proceso de correspondencia, que relaciona información en las diferentes cámaras, soporta el peso del algoritmo. No obstante, dependiendo de la configuración del sistema, ésta puede ser una tarea de excesiva dificultad y por tanto el sistema de múltiples cámaras no puede explotar las capacidades que posee.

- Método de extracción de características de la imagen: El método de extracción de características es compartido en el caso de una o múltiples vistas. Su misión es extraer características de la imagen que puedan ser identificadas unívocamente en distintas poses y que representen conceptos geométricos reconocibles mediante su proyección.
- Métodos de correspondencia:
 - Correspondencia temporal: Las marcas obtenidas en el plano imagen deben garantizar que son detectadas a lo largo de la evolución temporal que se observa en la imagen. Este tipo de procesos se ayudan de modelos de movimiento del objeto y de información previa existente en ese momento del tiempo.
 - Correspondencia espacial: En el caso de múltiples cámaras, se requieren métodos que aporten correspondencia entre cámaras alejadas, por lo que el problema es análogo al anterior si se suponen desplazamientos entre cada “frame” suficientemente alejados. Las técnicas usadas se especializan en aportar conocimiento de la calibración de cámaras y explotan las restricciones geométricas resultantes del esquema de múltiples vistas.

La división que se propone en esta sección diferencia en primer lugar los métodos de seguimiento basados en una o varias cámaras. Los métodos basados en el uso de una sola cámara se dividirán según la información previa aportada al sistema y estarán agrupados por el tipo de técnica de detección usada. En el caso de visión de múltiples cámaras, la división interna se centrará en los métodos de correspondencia propuestos.

2.4. Marcas naturales en visión monocular

El conocimiento previo, que un sistema basado en marcas naturales requiere, se presenta bajo diferentes formas. Puede provenir de un modelo CAD del objeto, un conjunto de planos conocidos o incluso modelos de poca precisión como un elipsoide que representa su volumen. El tipo de modelo determina las características buscadas, aportando un mayor grado de robustez cuanto mayor es la información que se tiene del objeto.

2.4.1. Métodos basados en modelo previo

Supuesto un modelo previo, se buscan aquellos métodos que sean capaces de encontrar puntos en la imagen y asociarlos a las primitivas (puntos, líneas, volúmenes) que definen el modelo. Una vez asociados, deberá ser capaz de calcular la correspondencia en imágenes sucesivas de tal modo que, aunando la información métrica del modelo con la correspondencia en imágenes consecutivas, se obtenga la posición y orientación del objeto del que se realiza el seguimiento.

Con el fin de organizar la extensa cantidad de trabajos que se enmarcan en este subconjunto, se propone una división basada en las características que se buscan en la imagen.

Métodos basados en detección de bordes

Los primeros trabajos propuestos en seguimiento de objetos están basados en detección de bordes; principalmente debido a que son computacionalmente eficientes y fácilmente implementables. Como añadido, los métodos basados en bordes presentan una gran inmunidad a cambios de iluminación global o gradual de la escena.

Se pueden agrupar en dos categorías:

- El primer enfoque se basa en buscar gradientes predominantes evitando su detección explícita.

Dentro de esta categoría se encuentra el método RAPiD [Harris, 1992], el cual se muestra en la figura 2.2, que fue uno de los primeros “tracker” 3D que funcionaban en tiempo real. La idea principal es considerar un conjunto de puntos 3D denominados puntos de control que se proyectarán sobre los gradientes predominantes de la imagen. El movimiento tridimensional del objeto entre dos frames consecutivos se recupera a partir de los desplazamientos en el plano imagen de la proyección de los puntos de control. La pose se resuelve mediante un algoritmo lineal de minimización, por lo que la carga computacional no es elevada.

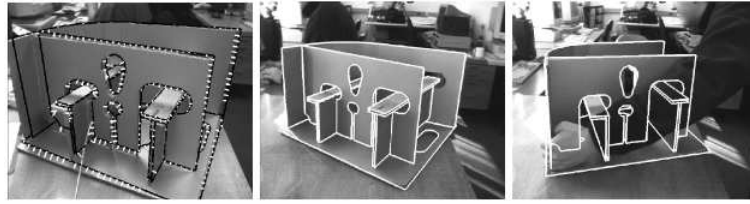


Figura 2.2: Métodos RAPiD basado en modelo

La aportación del método RAPiD se centra en que tan solo son necesarias las distancias perpendiculares a los contornos que se encuentran en la imagen (Ver figura 2.3). De esta manera, se realiza una correspondencia de un punto a un contorno, evitando la necesidad de proponer métodos que identifiquen el punto en sucesivas imágenes.

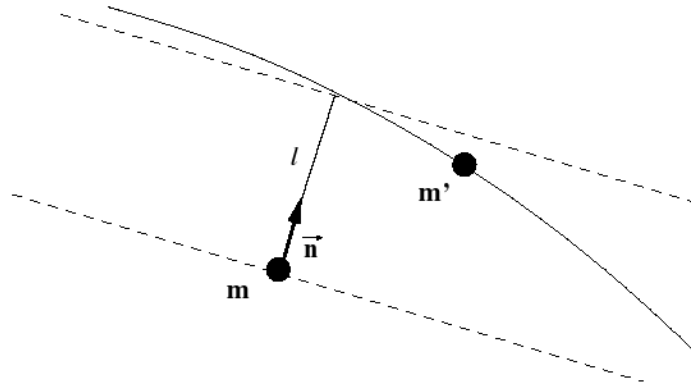


Figura 2.3: Distancia perpendicular a un contorno en RAPiD

El principal problema que presenta el método RAPiD es su falta de robustez. En numerosas ocasiones se asignan distancias a gradientes causados por sombras en el objeto, textura o ruido. En [Harris, 1992] se propone también un método que discrimina los contornos débiles y elimina el punto de control asociado aunque no elimina la problemática en casos reales.

Se han propuesto métodos para convertir RAPiD en un algoritmo estable mediante estimadores robustos [Drummond and Cipolla, 2002] [Marchand et al., 2001]. Esta propuesta requiere métodos de minimización no lineal y por tanto elimina parte de las ventajas del algoritmo. En [Armstrong and Zisserman, 1995] se propone el uso de un algoritmo RANSAC de detección de “outliers”.

- En el segundo caso, la imagen de gradiente se convierte en una imagen descrita mediante segmentos de recta que serán usados para realizar un ajuste con el modelo propuesto (ver figura 2.4).

Este enfoque ajusta por tanto primitivas geométricas del modelo , como pueden ser segmentos o curvas con las observadas en el plano imagen. Los trabajos mas relevantes se pueden encontrar en [Lowe, 1992] [Gennery, 1992] [Koller et al., 1993] [Kosaka and Nakazawa, 1995].

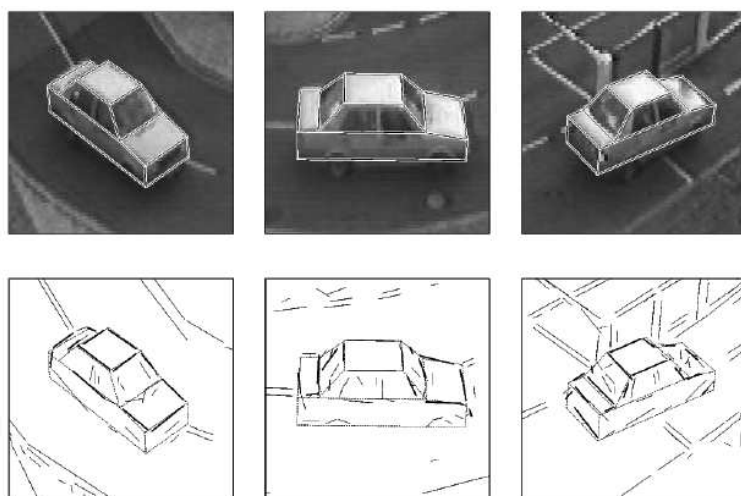


Figura 2.4: Métodos basados en ajuste de primitivas geométricas

En general las primitivas elegidas son segmentos de recta o curvas paramétricas sencillas. Cada uno de los segmentos de recta o curvas son proyectados en el plano imagen en función de la estimación de la pose del objeto. La correspondencia se resuelve mediante el empleo de un test de 'Mahalanobis' entre la estimación y las primitivas captadas en la imagen.

Muchas de las propuestas basadas en parametrizaciones han sido abandonadas debido a las restricciones geométricas a priori que imponen al objeto a detectar.

Métodos basados en flujo óptico

El flujo óptico se define como el movimiento aparente en el plano imagen resultado de la proyección de un punto físico en movimiento. Los métodos basados en flujo óptico parten de la suposición de que la intensidad de la proyección del punto permanece constante.

$$y' = y + \begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} dt \quad (2.1)$$

Donde y representa la proyección de un punto en la imagen $I(u, v, t)$ y y' representa su localización correspondiente en la próxima imagen $I(u', v', t + dt)$. Puesto que el flujo óptico se mide en el plano imagen, se define como un campo vectorial de dos dimensiones $(\dot{u}, \dot{v})^T$.

El método más utilizado fue propuesto por Lucas y Kanade [Lucas and Kanade, 1981], en el cual se adopta un enfoque multiescala para evitar el problema de apertura. En este método se supone suavidad en la variación del flujo óptico.

El uso de algoritmos basados en flujo óptico como único método de obtención de características y su correspondiente tracking fue propuesto desde sus inicios por [Li et al., 1993].

$$\left(\frac{\partial I}{\partial u}, \frac{\partial I}{\partial v}\right)(m' - m) + \frac{\partial I}{\partial t} = 0 \quad (2.2)$$

Este enfoque sin embargo adolece de carencias graves. En el caso de movimientos rápidos, la linealización que implica la ecuación (2.2) deja de cumplirse y la estimación puede alejarse del punto real. La solución iterativa de la anterior expresión por su parte produce una acumulación de error inevitable con efectos negativos en el “tracker”.

Para solventar el problema de la acumulación de error se proponen en [Li et al., 1993] técnicas basadas en síntesis recursiva. En este caso se aplican las ecuaciones de flujo óptico entre la nueva imagen capturada y una imagen sintética propuesta por un estimador de movimiento. Puesto que es prácticamente imposible tener en cuenta los cambios de iluminación en imágenes sintéticas, el sistema es muy sensible a ese factor. A su vez el hecho de necesitar varias imágenes sintéticas por cada frame lo hacen prohibitivo.

Otros enfoques incluyen algoritmos no lineales de minimización que tienen como objetivo la obtención del flujo óptico entre dos imágenes [Basu et al., 1996] [Black and Yacoob, 1997]. Estos enfoques pueden manejar movimientos más rápidos y aunque se pueden conseguir soluciones de baja carga computacional, están expuestos a error acumulativo.

Algunos autores combinan el flujo óptico de puntos con información de pertenencia a contornos. En [Haag and Nagel, 1999] se propone un Filtro de Kalman como elemento de fusión de información.

Un enfoque diferente se propone en [DeCarlo and Metaxas, 2000] donde se funde la información de pose, que aporta el conjunto de contornos detectados, con una restricción de flujo óptico estimado. La restricción se aplica directamente en la función de coste mediante un Lagrangiano, por lo que se debe resolver un problema de minimización sujeta a condiciones de contorno. El sistema permite incluso modelar el movimiento de objetos deformables. Las carencias que presenta este método tienen su origen en la suposición de constancia del brillo y la linealización formulada en (2.2). Todo esto supone que el sistema no es capaz de seguir movimientos rápidos ni de asimilar cambios de iluminación.

Métodos basados en ajuste a plantilla

A pesar de que el método de Lucas-Kanade [Lucas and Kanade, 1981] [Baker and Matthews, 2004] fue diseñado para el cálculo de flujo óptico, puede ser reformulado para el “tracking” de una plantilla bidimensional genérica ante una familia afín de deformaciones. Permite evitar centrarse en características locales, y utiliza toda la imagen correspondiente a la plantilla como fuente de información en el proceso de seguimiento.

A pesar de ser computacionalmente costoso, se puede demostrar [Hager and Belhumeur, 1998] que existen formulaciones alternativas que lo hacen tratable para el seguimiento actualizado. En varios trabajos se propone para el seguimiento de objetos tridimensionales. [Cascia et al., 2000] [Jurie and Dhome, 2001] [Jurie and Dhome, 2002].

El objetivo general del algoritmo de Lucas-Kanade consiste en encontrar los parámetros p que definen una transformación o ‘Warp’ $W(y; p)$ que traslada una plantilla T en la imagen de entrada I_t . La deformación $W(y; p)$ suele consistir en general en una transformación afín.

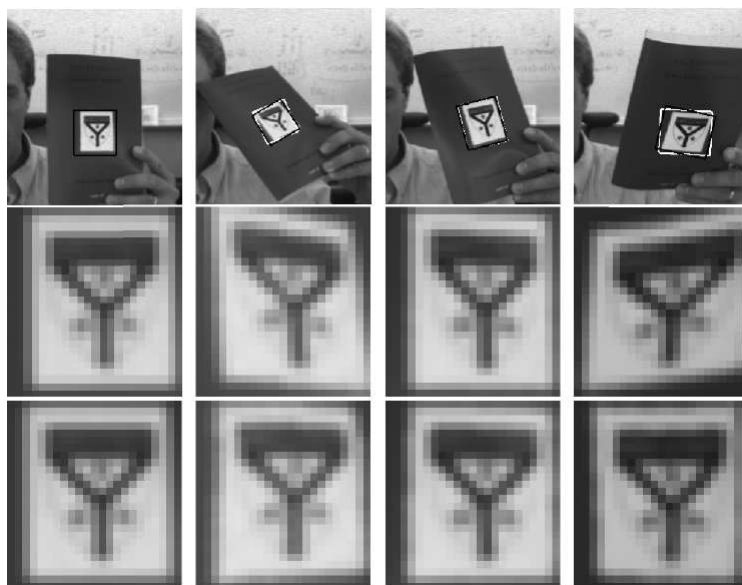


Figura 2.5: Método de ajuste a plantilla

$$\epsilon(p) = \sum_j (I_t(W(y_j; p)) - T(y_j))^2 \quad (2.3)$$

Los detalles de la formulación del algoritmo de Lucas-Kanade y una aproximación de la misma de baja carga computacional, conocida como el algoritmo composicional inverso, se describen en detalle en [Baker and Matthews, 2004].

Esta formulación, al igual que el flujo óptico, es poco robusta ante cambios de iluminación en la plantilla. En [Hager and Belhumeur, 1998] se propone una solución que hace uso de funciones de base que, mediante entrenamiento, son capaces de generalizar la textura de la plantilla ante cambios de iluminación. Otra desventaja de este método se hace visible cuando la plantilla está parcialmente ocluida. Este hecho puede dar lugar a una degeneración del método de optimización. En [Hager and Belhumeur, 1998] se propone también un método que utiliza estimadores robustos que ponderan cada píxel de la plantilla según el error de reproyección obtenido.

Una vez obtenida la plantilla es posible recuperar su pose mediante la homografía que define el plano que la forma y los parámetros intrínsecos de la cámara.

Métodos basados en extracción de puntos

En esta sección se comentarán métodos que hacen uso de marcas naturales puntuales en lugar de contornos o plantillas. Se basan en la detección de puntos individuales, lo cual permite dotar de inmunidad al algoritmo ante cambios de iluminación, oclusiones o errores de correspondencia. Este tipo de enfoque aprovecha más la información contenida en el plano imagen, permitiendo reducir las restricciones “a priori” en la apariencia del objeto.

Estos métodos están en cierta manera relacionados con los métodos basados en flujo óptico, sin embargo, puesto que no requieren pequeños desplazamientos en el plano imagen, pueden ser utilizados en caso de movimientos relativamente rápidos. Los métodos de tracking basados en extracción de puntos deben resolver dos problemas fundamentales:

- Identificación de aquellos puntos de la imagen que poseen buenas cualidades para ser seguidos en la imagen.
- Proponer un método de correspondencia temporal que relacione un punto en una imagen con el correspondiente en otra resultado del movimiento.

Discriminación de puntos de interés

La identificación de puntos de interés en la imagen posee una extensa bibliografía. En general se tiende a identificar cada punto con un conjunto de parches de textura recortados del plano imagen a lo largo del movimiento del objeto [Ravela et al., 1995] [Uenohara and Kanade, 1991]. A partir de esta técnica básica se proponen métodos para filtrar esa información y extraer propiedades que permitan identificar dicho punto cuando el objeto esté en una nueva posición.

La textura que rodea a un punto de interés sufre de deformaciones proyectivas, cambios de iluminación y ruido debido al fondo de la escena. En general no todos los puntos proyectados de la geometría de un objeto son susceptibles a ser seguidos en la imagen. Con el objetivo de discriminar aquellos puntos que garantizan un seguimiento estable del resto, se proponen en la literatura “operadores de interés” [Foerstner, 1986]. La propuesta de dichos operadores se remonta a las primeras publicaciones sobre “tracking” en los años 70 [Moravec, 1977] [Moravec, 1981] y hasta el presente se han continuado los trabajos en esta línea. En [Deriche and Faugeras, 1990] y [Smith and Brady, 1995].

Los operadores más utilizados, operador Förstner [Foerstner, 1986], Kanade-Lucas-Tomasi [Shi and Tomasi, 1994], o el famoso detector Harris [Harris and Stephens, 1988] se basan en el cálculo de la matriz de información de apariencia.

$$Z = \begin{pmatrix} \sum I_u^2 & \sum I_u I_v \\ \sum I_v I_u & \sum I_v^2 \end{pmatrix} \quad (2.4)$$

La matriz Z se calcula para todos los píxeles $I(u, v)$ de una región de la imagen identificada con un punto de interés. Los coeficientes I_u e I_v representan las derivadas con respecto a las coordenadas (u, v) de la imagen. En [Schmid and Mohr, 1997] se propone ponderar las derivadas usando ‘kernels’ gaussianos que penalizan aquellos puntos alejados del centro del parche de textura.

Mediante un análisis de autovalores de la matriz se discriminan aquellos puntos que no se comportan bien en el proceso de ‘tracking’. En general si ‘ Z ’ posee dos autovalores lo suficientemente grandes, el punto de interés se considera adecuado para su seguimiento.

Correspondencia entre imágenes

El proceso de correspondencia es un problema crítico en procesos de “tracking”. Para estimar el movimiento de un objeto se debe precisar, para el conjunto de puntos de interés $\mathcal{C}_k = \{y_k^1, \dots, y_k^{N_{C^n}}\}$ extraídos de la imagen $I(u, v, t)$, el conjunto de puntos correspondientes $\mathcal{C}_{k+1}$.

En [Zhang et al., 1995] se describe un método clásico y ampliamente aceptado que hace uso de un índice de correlación normalizado (ZNCC, “Zero Normalized Cross Correlation”) entre parches pertenecientes a puntos de interés en diferentes imágenes. La selección de la correspondencia está condicionada a que el máximo de correlación sea cruzado en ambos conjuntos de puntos. Es decir que la correspondencia asociada al punto y_k^i denotada como y_{k+1}^j tenga a su vez como máximo respectivo en la imagen origen al propio y_k^i . Otros métodos basados en enfoques parecidos se pueden encontrar en [Moravec, 1977] [Foerster, 1986], [Shi and Tomasi, 1994], [Simon et al., 2000], [Genc et al., 2002], [Chia et al., 2002], [Tordoff et al., 2002], [Li et al., 2004], [Vacchetti et al., 2004].

El éxito conseguido en este tipo de métodos depende en gran medida del ruido presente en la imagen y la deformación proyectiva. Para puntos pertenecientes a superficies planas y paralelas al plano imagen, se obtienen buenos resultados con este tipo de técnicas. Este tipo de soluciones suelen combinar métodos robustos como RANSAC para la eliminación de falsas correspondencias.

Una alternativa diferenciada es el uso del método de correspondencia de Kanade-Lucas-Tomasi [Lucas and Kanade, 1981], [Tomasi and Kanade, 1991], [Shi and Tomasi, 1994] que se basa en el propio flujo óptico para obtener la correspondencia. Este método es capaz de obtener muy buenos resultados en frames continuos y permite relajar las condiciones exigidas en la identificación de puntos de interés. Además, el número de falsas correspondencias es ínfimo. El principal problema de este enfoque es que es incapaz de aportar información de correspondencia cuando el espacio entre frames es grande. Las correspondencias mediante este método tienden a sobrevivir por muy poco tiempo. Al igual que en el flujo óptico, la correspondencia que se realiza para movimientos pequeños se ve interrumpida por oclusiones o cambios de aspecto.

Asociación con el modelo previo

Resueltos los problemas de la correspondencia en sucesivas imágenes, fruto del movimiento del objeto, se debe asociar el modelo con el conjunto de puntos de interés obtenidos en la imagen. En la mayoría de las aplicaciones se tiene una estimación de la pose inicial, la cual se usa para asociar en el plano imagen vértices del modelo proyectados, con puntos de interés detectados. Una vez solventado ese primer paso de inicialización, el proceso de seguimiento permitirá corregir la posición inicial haciendo uso de la posición en el plano imagen.

Existen muchos métodos propuestos para este fin, y que se pueden separar en dos vertientes:

- Métodos de obtención de pose mediante una transformación directa entre la posición

de los vértices proyectados y la posición en el plano de referencia de la cámara. Estos métodos incluyen el popular método POSIT [DeMenthon and Davis, 1995] y varias modificaciones equivalentes propuestas en [Christy and Horaud, 1999] [Liu et al., 1990].

- Métodos de obtención de pose mediante esquemas de corrección basados en la posición anterior y la posición actual de la proyección de los vértices del objeto. Este tipo de enfoque es más estable que el anterior y además posee la ventaja de que permite incluir posibles estimaciones del movimiento del objeto, favoreciendo un proceso de “tracking” robusto.

2.4.2. Métodos simultáneos de obtención del modelo

El conjunto de métodos presentados hasta el momento son capaces de estimar la posición dado un modelo tridimensional a priori. Sin embargo es posible la estimación simultánea del movimiento de la cámara y la geometría de la escena sin ningún modelo previo. La trayectoria recuperada y la posición de la cámara se referencian a un origen común, que por lo general corresponde a la posición inicial que ocupa la cámara.

Este problema conecta con una disciplina nueva y de gran crecimiento en los últimos años. Este proceso, conocido como SLAM o “*Simultaneous Localization and Mapping*”, surge ante la necesidad de posicionar un robot móvil, mediante sensores a bordo, en un entorno estructurado. Hasta la propuesta original de SLAM [Leonard and DurrantWhyte, 1991], las soluciones existentes consistían en obtener un mapa a priori supervisado del entorno mediante el conjunto de sensores. Una vez obtenido el mapa, la localización se resolvía identificando las observaciones realizadas con los sensores con características del mapa almacenado. Mediante este esquema es posible obtener la posición del robot.

Los procesos SLAM nacen con la intención de probar que es posible una solución en la que el mapa se construye al mismo tiempo que el robot realiza posicionamiento contra ese mapa dinámico. Dentro de este enfoque, el propio mapa se incluye dentro del conjunto de estados que se debe estimar en función de las medidas y un modelo de transición.

En general se proponen métodos de inferencia bayesiana que permitan resolver de manera secuencial el problema planteado. Los métodos SLAM han evolucionado en gran medida gracias a los buenos resultados que aportan. En [Newman and Leonard, 2003] [Montemerlo et al., 2002] [Knight et al., 2001] se presentan modificaciones que depuran las propuestas iniciales y que intentan solventar problemas de acumulación de error en grandes mapas e incluso fusión de mapas realizados mediante múltiples robots.

Dentro de esta disciplina, que a día de hoy acapara la atención de la comunidad dedicada al posicionamiento de robótica, se encuentran en pleno auge aquellos sistemas que utilizan la visión artificial como sensor base para completar el algoritmo SLAM. [Davison and Murray, 2002] [Davison, 2003] [Se et al., 2001]

El uso de visión artificial en un proceso SLAM supone en general resolver el problema de seguimiento clásico dentro de un enfoque probabilístico novedoso. De entre las soluciones planteadas, destaca el sistema propuesto en [Davison, 2002], en el cual se muestra un sistema SLAM mediante una cámara calibrada y su implementación en tiempo real. El

mapa que se realiza en este caso es poco denso y está al servicio del posicionamiento de la cámara de movimiento libre.

La aplicación de soluciones inspiradas en este tipo de métodos se encuentra en pleno crecimiento en la comunidad científica, tanto de robótica como de visión por computador. Las aplicaciones de realidad aumentada, debido a las soluciones existentes en tiempo real, han motivado su crecimiento. Del mismo modo, el enfoque probabilístico que ofrece, aporta más información que permite redefinir el proceso de correspondencia [Neira and Tardos, 2001].

La principal aportación de los sistemas SLAM a métodos de tracking consiste en su descripción probabilística que ayuda en la etapa de correspondencia. Otra ventaja que brinda el enfoque de SLAM es la posibilidad de construir modelos dinámicos a medida que el algoritmo se desarrolla.

La aplicación clásica trata de resolver el problema de posicionar una cámara mediante identificación de marcas en la imagen. Sin embargo su uso en el problema inverso de realizar tracking de objetos tridimensionales mediante una cámara es una solución de incipiente aplicación a sistemas que no están relacionados directamente con el problema inicial.

En [Pizarro et al., 2005] se muestra una aplicación de este tipo de enfoque para el posicionamiento de un robot móvil mediante un conjunto de cámaras externas.

2.5. Marcas naturales en visión de múltiples vistas

Como se ha comentado en el comienzo del capítulo, el problema que se plantea es el cálculo de la pose de un objeto rígido mediante un sistema de múltiples cámaras. La disposición de dichas cámaras atiende a un esquema “wide-baseline”, es decir las cámaras presentan centros de proyección alejados. En este esquema el proceso de correspondencia se complica debido a la poca correlación que tienen las distintas imágenes de un mismo objeto.

Los procesos que se han visto en este capítulo, los cuales se aplican a la detección de marcas naturales en visión monocular, basan su principio de funcionamiento en la correlación existente en diferentes momentos de tiempo. Si el desplazamiento del objeto es suficientemente grande, el problema, en configuración de una sola cámara también se convierte en uno de “wide-baseline” y por tanto los métodos que se exponen a continuación son igualmente recomendables.

En el caso de múltiples cámaras, si el proceso de correspondencia se resuelve de manera satisfactoria, el modelo se obtiene de manera automática. En este caso, por tanto, no existe la diferenciación en cuanto a la información a priori, independientemente de que ésta pudiera ser útil para ayudar al proceso de correspondencia.

El origen de los métodos de extracción de marcas naturales que sean reconocibles ante deformaciones proyectivas fuertes proviene de la disciplina de detección de objetos en la imagen basándose en su apariencia. Este tipo de enfoque propone la detección de objetos en imágenes, empleando una base de datos de características obtenidas a partir de un conjunto de imágenes de entrenamiento [Nayar et al., 1996], [Viola and Jones, 2001].

El objetivo es encontrar una descripción del objeto que sea invariante a cambios proyectivos y fácilmente formulable e identificable en el plano imagen. A su vez debe ser estable ante cambios de iluminación.

En [Lindeberg, 1994] se propone un descriptor invariante a la escala, mediante la obtención de los máximos locales de un espacio piramidal definido por el filtro laplaciano de gaussiana. Con el objetivo de disminuir la complejidad del algoritmo se propuso aproximar dicho filtro mediante una diferencia de gaussianas escaladas (Difference of Gaussians D.O.G). A partir de dicho espacio piramidal, que ofrece invariancia a la escala, la investigación se ha centrado en encontrar una descripción invariante a la proyección. En muchos casos [Baumberg, 2000] [Schaffalitzky and Zisserman, 2002] [Mikolajczyk and Schmid, 2002] se proponen variantes del detector “Harris” donde la región equivalente se rectifica y normaliza en función de los autovalores de la matriz de apariencia. En [Vacchetti et al., 2004] [Matas et al., 2002] y [Tuytelaars and VanGool, 2000] se proponen métodos que también se basan en propiedades de la textura en el plano imagen, lo que los hace menos inmunes a los cambios de iluminación.

2.5.1. Métodos basados en descriptores invariantes

Un descriptor local busca extraer información de un área localizada de la imagen, como puede ser la textura alrededor de un punto de interés, y define un método de codificación que sea invariante a cambios de iluminación y transformaciones afines. Los descriptores deben ser definidos de tal modo que a su vez permitan separar de manera apreciable dos áreas que no pertenecen al mismo punto.

La literatura es extensa en el número de descriptores que se proponen. [Baumberg, 2000], [Tuytelaars and VanGool, 2000] [Mikolajczyk and Schmid, 2002] [Schaffalitzky and Zisserman, 2002]. Mediante este tipo de métodos es posible detectar marcas en una superficie plana ante cambios de posición y orientación de la superficie, supuesta una cámara. Sin embargo la invariancia afín completa no se consigue puesto que la elección de las características se realiza mediante métodos que no son totalmente invariantes debido al coste computacional que eso requeriría.

En [Mikolajczyk and Schmid, 2002] se demuestra que la calidad en el tracking de dichas marcas invariantes es inferior que aquellas basadas simplemente en invariancia a la escala. En añadido, los métodos que intentan aportar invariancia afín completa, son menos inmunes al ruido y pueden ser muy restrictivos en el caso de marcas no planares.

Método SIFT (Scale-Invariant Keypoint Transform)

El método SIFT propuesto originalmente por Lowe [Lowe, 1999], es hasta la fecha la solución propuesta con más éxito en entornos reales. Las marcas que detecta el método SIFT son invariantes ante cambios de escala, rotación y parcialmente inmunes a cambios de iluminación y deformación proyectiva. Poseen propiedades de localización tiempo-frecuencia buenas y ofrecen descriptores que permiten ser distinguidos en imágenes reales.

Mediante este método se demuestra experimentalmente que se pueden extraer un gran

número de marcas mediante algoritmos eficientes. A su vez los descriptores permiten identificar marcas a lo largo del movimiento de un objeto o comparar marcas extraídas de diferentes poses en bases de datos para identificación.

El método SIFT se compone de los siguientes pasos:

- **Detección de máximos en el espacio de escala.** El primer paso es obtener un espacio piramidal de diferencias de gaussianas mediante un banco de filtros (figura 2.6). Mediante este espacio se evalúan aquellas localizaciones en escala y posición que son susceptibles de ser un punto de interés. Dichos puntos coinciden con los máximos y mínimos de ese espacio.

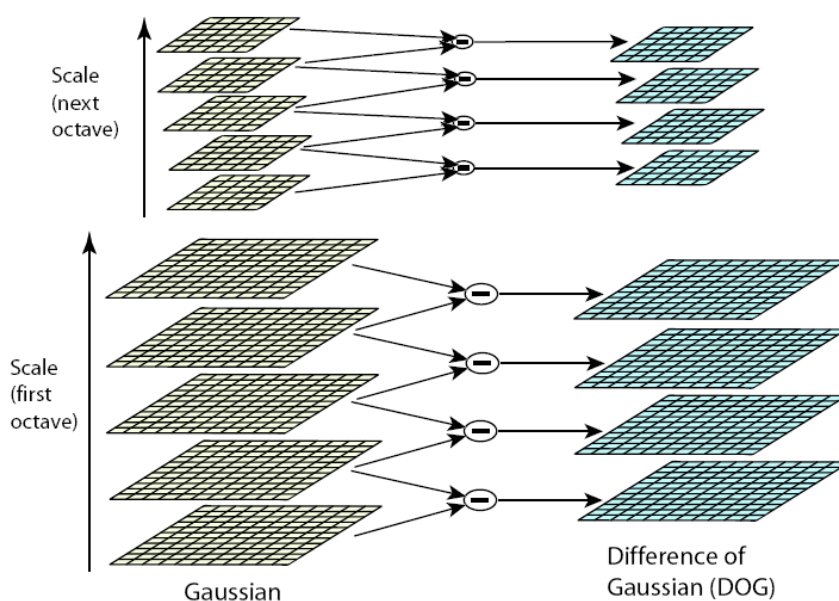


Figura 2.6: Espacio de diferencias de gaussianas utilizado en el método SIFT

- **Localización de los puntos de interés.** En cada punto candidato de ser un punto de interés se realiza un ajuste de modelo que permite encontrar con precisión la escala y la posición del punto. Para ello se realiza una aproximación de función cuadrática del espacio de escala local al punto detectado (ver figura 2.7), lo cual permite buscar un valor refinado del máximo. El método SIFT propone a su vez una medida de estabilidad que descarta aquellos candidatos que no son susceptibles de ofrecer un descriptor distinguible.
- **Obtención de la orientación local:** Una o varias orientaciones principales son asignadas a cada punto clave obtenido en el proceso anterior. Dicha orientación se obtiene a través de un histograma de direcciones de gradientes en el espacio imagen. Una vez asignadas, todas las operaciones posteriores se referencian a la escala y orientación calculadas. La invariancia a la rotación y escala se obtiene en este punto.
- **Descriptor.** Los gradientes locales de la imagen, referentes a la escala y la rotación calculada, se utilizan para construir un descriptor que posee inmunidad ante distor-

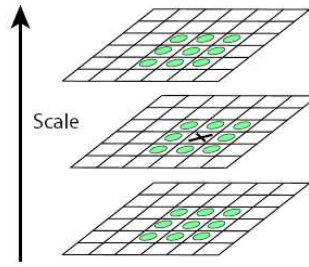


Figura 2.7: Detección de máximos

siones y cambios de iluminación. Se compone de pequeños histogramas de orientación localizados en subventanas del área de interés del punto (figura 2.8). El resultado es un descriptor que generalmente se compone de 128 componentes que se compara en sucesivas imágenes mediante distancia euclídea.

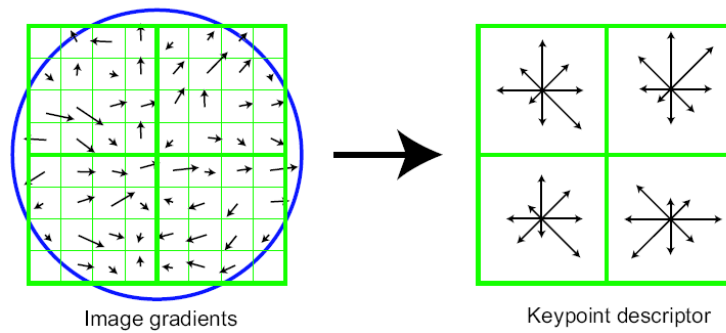


Figura 2.8: Descriptor propuesto en el método SIFT

El método SIFT es ampliamente aceptado y utilizado gracias al buen comportamiento que presenta en procesos de tracking y reconocimiento de objetos. La razón de los buenos resultados que presenta se encuentra en la definición del descriptor. Dicho descriptor permite pequeños desplazamientos y se inspira en los estudios realizados en el procesamiento que se realiza en el ojo humano para la detección de puntos de interés.

El autor demuestra en sucesivos trabajos [Se et al., 2001] [Lowe, 2004] [Lowe, 2001] su aplicación a realidad aumentada, reconocimiento de objetos en bases de datos de imágenes y en técnicas SLAM.

Se han propuesto alternativas y modificaciones del algoritmo SIFT que permiten mejorar su eficacia. En [Stein and Hebert, 2005] se propone un método que permite eliminar la influencia de las variaciones de fondo en marcas que están en el borde de un objeto.

El método PCA-SIFT [Ke and Sukthankar, 2004] se basa en una técnica de análisis de componentes principales para la determinación del descriptor que define al objeto en lugar de los histogramas. Se demuestra que dicho método produce descriptores más compactos, robustos y distinguibles entre puntos de la escena.

Métodos SIFT en tracking 3D monocular

En [Skrypnyk and Lowe, 2004] se propone el uso del método SIFT para realizar el tracking 3D de objetos. El uso de este tipo de descriptores en este caso parte de una filosofía distinta al resto de técnicas comentadas en visión monocular. En este caso se explotan las propiedades de detección que presenta los métodos “wide-baseline”.

En primer lugar, durante un proceso de inicialización, se obtiene un conjunto de correspondencias a través de múltiples imágenes. Mediante una técnica exproceso de reconstrucción a partir del movimiento, se determinan las coordenadas tridimensionales de los descriptores obtenidos, así como los parámetros de la cámara.

Una vez que existe un modelo del mundo, la posición y orientación de la cámara se obtiene, para cada instante de tiempo, mediante el establecimiento de la correspondencia de descriptores. La búsqueda en la base de datos inicialmente construida es un problema de coste computacional alto. En [Beis and Lowe, 1997] se propone un método que permite reducir la complejidad y obtener la correspondencia con alta probabilidad.

La pose de la cámara se obtiene mediante un método RANSAC para evitar la aparición de “outliers”. Finalmente se aplica una condición de regularización al movimiento de la cámara que robustece el algoritmo.

En esta propuesta, a diferencia de los métodos SLAM, una vez obtenido el modelo previo de la escena, la pose de la cámara se obtiene de manera absoluta para cada frame. La posición actual no depende de la estimación anterior, lo que permite eliminar el posible error de acumulación existente en otros métodos.

A pesar de tener un coste computacional alto, se encuentran soluciones capaces de operar a varias imágenes por segundo. El principal problema derivado de este tipo de métodos es que tienden a ser menos precisos que sus respectivos recursivos. Los descriptores SIFT poseen menor localidad espacial y requieren de la imposición de restricciones de movimiento continuo.

2.6. Conclusiones

En este capítulo se ha expuesto una revisión de las aportaciones realizadas en el seguimiento de objetos rígidos mediante visión artificial.

El objetivo propuesto en esta tesis se traslada fácilmente a un problema de “tracking” mediante cámaras, cuya configuración “wide-baseline” impone restricciones severas. Como añadido, se pretende eliminar el conocimiento a priori que el sistema de visión posee sobre el objeto.

A modo de resumen, en la figura 2.9 se muestran las diferentes técnicas utilizadas en el problema de “tracking” mediante visión artificial. La división establecida en la figura 2.9 comienza por diferenciar métodos que hacen uso de marcas artificiales con aquellos que utilizan marcas naturales. En este último grupo, el cual es de interés para la presente tesis, se diferencian aquellos basados en la utilización de una única vista y aquellos que tienen en cuenta un conjunto de cámaras. El capítulo se ha ordenado de modo que las

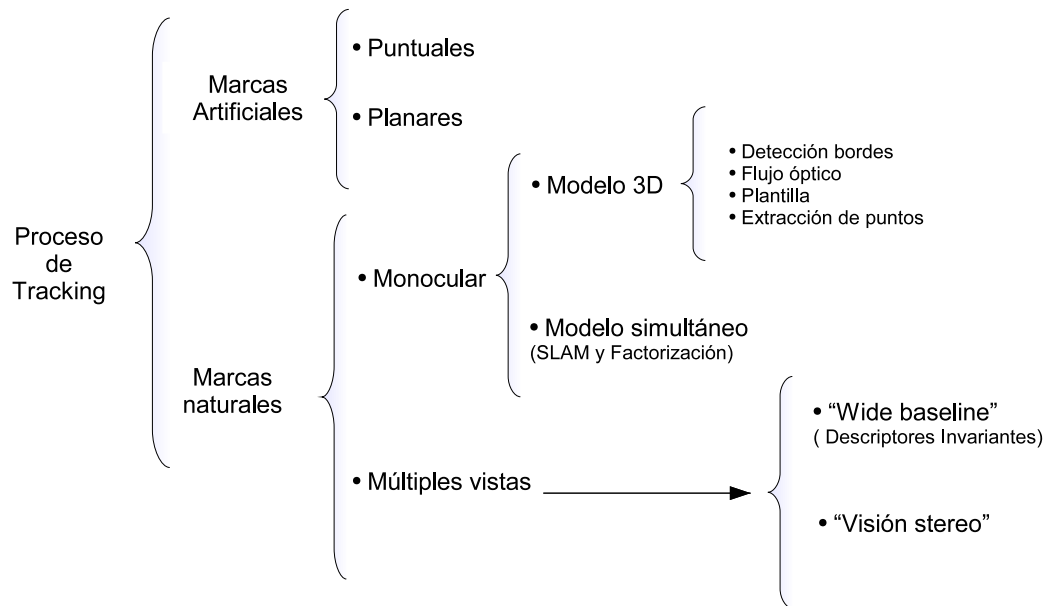


Figura 2.9: Organización de las técnicas de tracking mediante visión artificial

técnicas decrecen en exigencias de conocimiento previo acerca del objeto del que se realiza el tracking.

Los métodos de tracking comentados varían en resultados y escenarios de aplicación. A continuación se presenta un resumen de sus características principales:

- Métodos de detección de marcas artificiales: El conocimiento previo aportado al objeto del que se realiza el seguimiento tiene como recompensa un método robusto y rápido aplicable a cualquier configuración de cámaras. El sistema sin embargo no tiene interpretación del entorno y se convierte en un proceso fotométrico.
- Métodos de detección de marcas naturales: Dentro de este tipo, se diferencian aquellos basados en la información proveniente de una o múltiples cámaras.

- Métodos de tracking 3D mediante una cámara: Este tipo de métodos tiene a su vez una división según la información a priori necesaria.

Aportando un modelo 3D inicial, se han presentado propuestas que permiten extraer características en el plano imagen y realizan una comparación con la proyección del modelo. La variedad de estos enfoques, tiene como resultado diferentes comportamientos. Los métodos basados en primitivas geométricas, contorno o plantillas poseen en general buenas propiedades de precisión pero se encuentran serias dificultades en inmunizarlos ante oclusiones o cambios de iluminación en el caso de plantillas. Los métodos basados en flujo óptico no permiten movimientos bruscos y los cambios de iluminación destruyen la suposición de constancia del brillo. Sin embargo tienen muy buenas propiedades en

la asignación de correspondencias. Como contrapartida, los métodos de extracción de puntos, permiten cambios de iluminación y movimientos más rápidos. El inconveniente se presenta en el proceso de correspondencia.

En el caso de no aportar un modelo tridimensional, las técnicas SLAM apuestan por una solución ambiciosa que propone construir el modelo al tiempo que se detecta o posiciona el objeto. Mediante SLAM es posible redefinir el proceso de correspondencia de manera que mediante técnicas como los puntos de interés anteriormente comentados, se soluciona parcialmente el problema de la detección de marcas naturales. El problema significativo en este tipo de métodos es el error acumulativo que presentan en el momento de añadir nuevas marcas al vector de estados. Además el cálculo de correspondencias que definen en un marco probabilístico depende en gran medida de la calidad del proceso de estimación. Como conclusión a este tipo de técnicas, se plantea la necesidad de un método externo de apoyo que permita la corrección de errores acumulativos y fallos de detección.

- Métodos de tracking mediante múltiples cámaras: El tipo de configuración “wide-baseline” que se propone en un espacio inteligente, restringe la utilización simultánea de cámaras a técnicas que sean capaces de trabajar en un entorno en el que la deformación proyectiva es severa. Este tipo de técnicas no aseguran una cantidad elevada de correspondencias correctas cuando la separación entre las cámaras es significativa, aumentando considerablemente su complejidad.

Sin embargo, los métodos “wide-baseline” presentan buenas cualidades debido a su inmunidad frente a cambios de iluminación y perspectiva. El detector SIFT ha demostrado ser una herramienta indispensable para solucionar el problema de correspondencia planteado, incluyendo aquellos casos con una sola cámara en los que el objeto se ha desplazado lo suficiente.

En la presente tesis se propondrá la sinergia entre diferentes métodos de “tracking” y procesos de medida en el plano imagen, con la intención de obtener un método que sea posible utilizar indistintamente en configuraciones de una o múltiples cámaras.

Capítulo 3

Estructura y modelado matemático de los elementos del sistema

3.1. Introducción

El presente capítulo tiene como objetivo el modelado matemático de todos los procesos involucrados en el sistema de localización propuesto en esta tesis, con el fin de servir como referencia en capítulos posteriores y base para la consecución de los objetivos planteados.

El “Espacio Inteligente” está formado por elementos dinámicos, como pueden ser robots móviles o usuarios, y elementos sensoriales, como las cámaras o sensores de odometría. En particular, se confeccionará un modelo matemático de los siguientes elementos, involucrados directamente en obtener la pose del robot:

- Disposición y modelo de proyección en la imagen de un conjunto de cámaras.
- Estructura tridimensional de un objeto rígido y modelos de movimiento del mismo.
- Descripción matemática de las incertidumbres debidas a ruido en las magnitudes medidas e imprecisiones de los modelos matemáticos

La notación establecida para el transcurso del documento se muestra en la Tabla 3.1

3.2. Robot móvil

Cada uno de los robots, de los cuales se pretende obtener su localización, forman parte integrante del “Espacio Inteligente”, y por tanto sus sensores internos y su movimiento son supervisados remotamente a través de una red de comunicaciones. Dicha red se considera transparente para el sistema, y los posibles efectos relativos al enlace de comunicaciones existente no se consideran en el planteamiento matemático de la tesis.

Todos los robots controlados por el “Espacio Inteligente” disponen de sensores de odometría, los cuales ofrecen una estimación relativa de la pose. Dicha información será de especial interés en las soluciones marginadas al uso de una sola cámara. La odometría presente en el robot está sincronizada y disponible junto con la adquisición de imágenes y por lo tanto es posible relacionar ambas magnitudes en un índice temporal discreto.

El objetivo principal de esta tesis es obtener la posición y orientación de cada uno de los robots móviles presentes en el espacio de observación de las cámaras. Para ello se propone un método que relaciona la pose de un objeto y su apariencia en el plano imagen del conjunto de cámaras. Suponiendo que todos los robots son objetos rígidos, dicha relación se sustenta mediante la obtención de un modelo tridimensional del robot cuya apariencia en la imagen sea fácilmente discriminable (e.g. puntos, líneas, regiones). De este modo, dada la identificación del modelo en la imagen, la pose es obtenida mediante un sistema que resuelve un problema de perspectiva inversa en una o varias cámaras.

En el presente apartado, no solo el modelo de movimiento del robot presenta importancia, sino también el modelo geométrico que relacionará la estructura del robot con la imagen captada por la cámara.

Se utilizarán letras mayúsculas para vectores y matrices. Por definición, los vectores se expresan en formato columna.

- $k \rightarrow$ índice temporal discreto, global para todas las magnitudes.
- $k^1, \dots, k^2$, intervalo consecutivo de tiempo.
- $O_W \rightarrow$ origen de coordenadas global o del mundo.
- $O_{C_n} \rightarrow$ Sistema de coordenadas de la cámara n , que coincide con su origen de proyección.
- $O_R \rightarrow$ origen de coordenadas del robot.
- x^1, x^2, x^3 denotan las coordenadas tridimensionales del espacio cartesiano.
- $X_k \rightarrow$ vector que codifica la pose del robot (posición y orientación), referenciado a O_W .
- $M^1, \dots, M^N \rightarrow$ conjunto de N puntos tridimensionales que definen la estructura o modelo geométrico del robot y referenciados a O_R .
- $M_{X_k}^i \rightarrow$ i ésimo punto tridimensional de la estructura del robot referenciado a O_W , mediante la pose actual X_k .
- $X_k^a \rightarrow$ vector conjunto o aumentado que contiene la pose del robot y los puntos de su estructura.
- $y_k^{i,n} \rightarrow$ i ésima medida bidimensional en la cámara n en el instante k . $y_k^{i,n} = (u_k^{i,n}, v_k^{i,n})^T$.
- $C_k^n \rightarrow$ conjunto de medidas en el instante k en la cámara n . $C = \{y^{1,n}, \dots, y^{N_n,n}\}$.
- $N_{C^n} \rightarrow$ número de medidas en el instante k tomadas por la cámara n .
- $Y_k^n \rightarrow$ vector columna que contiene todas las medidas del conjunto C_k^n , que corresponden a puntos del modelo del robot.
- $N_n \rightarrow$ número de medidas en el instante k , tomadas por la cámara n que corresponden a puntos de la estructura del robot.
- $Y_k \rightarrow$ vector columna que contiene las medidas correspondientes a la estructura del robot en todas las cámaras disponibles.
- $U_k \rightarrow$ vector con lecturas de odometría $(Vl_k \quad \Omega_k)^T$ que consisten en velocidad lineal y angular respectivamente.

Tabla 3.1: Símbolos matemáticos y notación utilizada

3.2.1. Pose del robot

Se define un sistema de coordenadas O_R solidario con el volumen ocupado por el robot. La pose del robot describe la orientación y el desplazamiento existente entre dicho sistema de coordenadas y el correspondiente al mundo O_W , el cual es definido en una etapa previa. En el instante de tiempo k , se define X_k como el vector que describe la pose de un robot concreto. Para caracterizar la pose del robot en el espacio tridimensional, el vector X_k se compone de 3 coordenadas x_k^1, x_k^2, x_k^3 para el desplazamiento y 3 para los ángulos de Euler $\gamma_k, \beta_k, \alpha_k$ que definen la rotación en el espacio (ver Fig. 3.1).

$$X_k = (x_k^1, x_k^2, x_k^3, \alpha_k, \beta_k, \gamma_k), \quad (3.1)$$

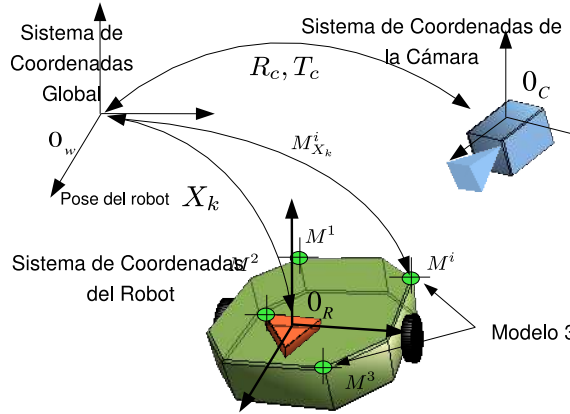


Figura 3.1: Relación espacial entre el sistemas de coordenadas global O_W , el sistema de coordenadas del robot O_R y el sistema de coordenadas O_C de una cámara cualquiera.

Los ángulos de Euler en el caso tridimensional presentan problemas conocidos de estabilidad. En su lugar es preferible, como una solución más estable para expresar rotaciones en 3 dimensiones, el empleo de los denominados cuaterniones unitarios $q_k = \{q_k^1, q_k^2, q_k^3, q_k^4\}$, donde $|q_k| = 1$. El principal inconveniente en la utilización de cuaterniones unitarios se traduce en la necesidad de introducir una dimensión extra, con respecto a los 3 ángulos de Euler, para expresar la rotación:

$$X_k = (x_k^1, x_k^2, x_k^3, q_k^1, q_k^2, q_k^3, q_k^4) \quad |q_k| = 1, \quad (3.2)$$

No obstante, en la mayoría de las aplicaciones que conciernen la localización de robots móviles en espacios interiores, se considera que el movimiento del robot se encuentra restringido a un plano. En dichos casos la pose del robot se reduce a la posición en un plano de referencia $(x^1 - x^2)$ (Plano X-Y cartesiano) y orientación α sobre el eje x^3 , formando un vector de pose de 3 dimensiones $X_k = (x_k^1, x_k^2, \alpha_k)$.

3.2.2. Modelos de movimiento del robot

En la presente tesis se supone la existencia de un modelo de movimiento del robot en cuestión, ya sea en el espacio tridimensional como en el plano, y que se encapsula en la función $X_k = g(X_{k-1}, U_k)$. Se supone un modelo de movimiento relativo, donde la función g requiere la posición anterior a la actual X_{k-1} y un vector de entrada U_k donde se encuentra contenida la información proporcionada por los sensores de odometría. En general el vector U_k describe incrementos de posición o equivalentemente velocidades o aceleraciones del robot, y por tanto la pose anterior es necesaria para obtener una estimación referenciada desde el sistema global O_W .

El contenido del vector U_k variará en función del modelo de movimiento usado por el robot. En este documento se referirá a dicho contenido como odometría en general, pero igualmente es susceptible de incluir lecturas de sensores de aceleración, inclinación, y demás valores necesarios en cada tipo de plataforma para obtener una estimación de la pose.

Las plataformas robóticas reales que circularán por el “Espacio Inteligente” y de las que se calcula la pose, consistirán en robots con ruedas cuyo movimiento está restringido a un plano. De esta modo, muchas de las conclusiones estarán específicamente dispuestas para este tipo de movimiento y será por tanto el utilizado en todos los experimentos.

De entre los tipos de robots que se mueven en un plano adoptaremos el modelo de tipo no-holonómico que toma como vector U_k las velocidades angulares y lineales del robot. Dicho modelo es suficientemente genérico como para estimar la pose de diferentes plataformas robóticas reales.

El modelo de movimiento considerado será por tanto:

$$\begin{pmatrix} x_k^1 \\ x_k^2 \\ \alpha_k \end{pmatrix} = \begin{pmatrix} x_{k-1}^1 + V l_k \cos(\alpha_{k-1} + \Omega_k) \\ x_{k-1}^2 + V l_k \sin(\alpha_{k-1} + \Omega_k) \\ \alpha_{k-1} + \Omega_k \end{pmatrix}, \quad (3.3)$$

donde $V l_k$ y Ω_k representarán respectivamente las velocidad lineal y angular del robot en el instante de tiempo k . Ambas magnitudes son agrupadas por tanto en el vector de entrada $U_k = (V l_k, \Omega_k)^T$. Mediante una regla recursiva de estimación, que parte de un vector inicial X_0 de posición, y aplica el modelo de movimiento descrito por (3.3), se obtiene una estimación de la posición y orientación X_k para cualquier instante de tiempo.

La estimación de movimiento usando odometria o “Dead Reckoning” presenta inconvenientes bien documentados en la literatura y que derivan en una estimación de la pose del objeto que presenta un error acumulativo con la longitud del trayecto. Este tipo de error será modelado con detalle en esta tesis y jugará un papel importante en la etapa de inicialización del sistema.

3.2.3. Modelo geométrico de la estructura del robot

La geometría del robot se modela como un conjunto de N puntos tridimensionales agrupados en $M = \{M^1, \dots, M^N\}$ y referenciados desde el sistema de coordenadas del robot O_R . La suposición de rigidez en la geometría permite afirmar que los puntos agrupados en

M son invariantes en el tiempo y por tanto se evitará el uso de subíndices temporales en ellos. Mediante el uso de la función $M_{X_k}^i = t(X_k, M^i)$ es posible trasladar debidamente un punto perteneciente a la estructura del robot y representarlo referenciado con respecto a O_W . Por ejemplo, para un robot que se mueve en un plano, la función $t(X_k, M^i)$ será:

$$M_{X_k}^i = t(X_k, M^i) = \begin{pmatrix} \cos(\alpha_k) & \sin(\alpha_k) & 0 \\ -\sin(\alpha_k) & \cos(\alpha_k) & 0 \\ 0 & 0 & 1 \end{pmatrix} M^i + \begin{pmatrix} x_k^1 \\ x_k^2 \\ 0 \end{pmatrix} \quad (3.4)$$

En el caso de movimiento tridimensional, la transformación $t(X_k, M^i)$ consiste en una transformación afín, compuesta por una rotación genérica R_k y un desplazamiento T_k en las tres dimensiones espaciales.

La obtención del punto M^i en el sistema de coordenadas del mundo O_W requiere conocer la pose actual del robot y por tanto el punto transformado $M_{X_k}^i$ tiene una dependencia con la pose y consecuentemente un índice temporal. La importancia de $M_{X_k}^i$ reside en que permite establecer la relación existente entre la proyección en el plano imagen $y_k^i = h(M_{X_k}^i, P)$ de un punto M^i del robot con la pose X_k .

Con el objetivo de agrupar toda la información útil que el sistema posee sobre el robot, se define el vector de estados aumentado X_k^a como la concatenación en un solo vector columna de la pose del robot X_k (ya sea en espacio tridimensional o en el plano) y el conjunto de puntos pertenecientes a su estructura geométrica M .

$$X_k^a = \begin{pmatrix} X_k \\ M^1 \\ \vdots \\ M^N \end{pmatrix} \quad (3.5)$$

El vector aumentado representa la solución de la pose del robot y la geometría conocida del mismo en un instante de tiempo dado.

Un modelo equivalente de movimiento $X_k^a = g^a(X_{k-1}^a, U_k)$ se define para el vector aumentado. Para formar g^a se utiliza el modelo de movimiento anteriormente comentado y la suposición de geometría rígida en el robot:

$$g^a(X_{k-1}^a, U_k) = \begin{pmatrix} g(X_{k-1}, U_k) \\ M^1 \\ \vdots \\ M^N \end{pmatrix} \quad (3.6)$$

3.3. Modelo y distribución de cámaras

Un conjunto de N_c cámaras es utilizado como red de sensores por el “Espacio Inteligente” con la cual obtener la posición y orientación de los robots visibles por al menos una de ellas.

El conjunto de cámaras presenta una disposición geométrica conocida a priori con suficiente precisión, mediante la realización de un proceso de calibración. Cada una de las cámaras se encuentra conectada a una red local, de manera que la adquisición de imágenes sea accesible por los sistemas de procesamiento. La topología física utilizada, así como las tecnologías de comunicaciones involucradas serán detalladas en un capítulo dedicado a describir la fase experimental de la tesis. Desde el punto de vista teórico, cabe destacar que la adquisición se considera estrictamente sincronizada entre el conjunto de cámaras y los parámetros extrínsecos e intrínsecos de cada cámara son conocidos a priori. A continuación, y a modo de referencia se comentará brevemente el modelo de cámara y la notación utilizada.

3.3.1. Modelo de cámara

Para una cámara cualquiera n , dispuesta en el espacio inteligente, se tiene un modelo de proyección genérico $y = h(M_X, P^n)$, el cual transforma un punto tridimensional $M_X = (x^1, x^2, x^3)^T$, referenciado desde el origen de coordenadas global O_W , en su proyección bidimensional $y = (u, v)^T$ en el plano de la cámara. La transformación desde tres dimensiones al plano imagen es gobernada por un conjunto de parámetros de proyección de cámara P^n que encapsulan tanto parámetros extrínsecos o puramente geométricos y los parámetros intrínsecos de la misma.

Existen un conjunto amplio de posibles modelos de cámaras, definidos por sus respectivas parametrizaciones. En la práctica, la mayoría de las cámaras usadas en aplicaciones similares a la que se presenta en esta tesis son descritas por un modelo “pin-hole” simple de proyección.

El vector de parámetros en dicho caso es el siguiente:

$$P^n = (\alpha_c^n \quad \beta_c^n \quad \gamma_c^n \quad T_c^n \quad f_u^n \quad f_v^n \quad u_0^n \quad v_0^n)^T, \quad (3.7)$$

donde $(\alpha_c^n \quad \beta_c^n \quad \gamma_c^n)$ y T_c^n , representan respectivamente los ángulos de rotación de Euler y el vector de traslación en el espacio del sistema de coordenadas de la cámara O_C^n con respecto al sistema de coordenadas global O_W . Mediante los mismos, la matriz afín de transformación de coordenadas espaciales entre ambos sistemas de coordenadas se forma como una matriz de rotación $R_c^n(\beta_c^n, \gamma_c^n, \alpha_c^n)$ y el vector de traslación T_c^n . Los parámetros f_u^n, f_v^n, u_0^n y v_0^n representan el modelo de parámetros intrínsecos simplificado, en el cual se supone que el modelo de cámara es “pin-hole” ideal sin parámetro de “skew” ni parámetros de distorsión. Los parámetros intrínsecos se aglutinan en la matriz K_c^n , la cual se describe a continuación:

$$K_c^n = \begin{pmatrix} f_u^n & 0 & u_0^n \\ 0 & f_v^n & v_0^n \\ 0 & 0 & 1 \end{pmatrix} \quad (3.8)$$

La transformación no homogénea resultante h para el modelo convencional “pin-hole” queda por tanto definido, para un punto genérico M_X y un vector de parámetros P^n como:

$$h(M_X, P^n) = \begin{pmatrix} f_u^n \frac{R_{1,1}^n x^1 + R_{1,2}^n x^2 + R_{1,3}^n x^3 + T_1^n}{R_{3,1}^n x^1 + R_{3,2}^n x^2 + R_{3,3}^n x^3 + T_3^n} + u_0^n \\ f_v^n \frac{R_{2,1}^n x^1 + R_{2,2}^n x^2 + R_{2,3}^n x^3 + T_2^n}{R_{3,1}^n x^1 + R_{3,2}^n x^2 + R_{3,3}^n x^3 + T_3^n} + v_0^n \end{pmatrix}, \quad (3.9)$$

donde $R_{i,j}^n$ representa el elemento i, j de la matriz de rotación R_c^n y el escalar T_i^n representa el elemento i del vector T_c^n .

En una sección posterior se analizará la influencia de los errores producidos en la calibración en los modelos de cámara utilizados.

Cada una de las N_c cámaras de las que se dispone, posee un vector P^n de parámetros de calibración diferente del resto, el cual es obtenido mediante un proceso de calibración previo. El vector columna conjunto $P = ((P^1)^T, \dots, (P^{N_c})^T)^T$ engloba todos los vectores de parámetros para las N_c cámaras disponibles, representando la calibración completa del “Espacio Inteligente”.

3.3.2. Vector de medidas en el plano imagen

El conjunto de medidas tomadas por una cámara cualquiera n , corresponde con la proyección en el plano imagen de los puntos pertenecientes a la geometría conocida del robot y encapsulados en X_k^a junto con la pose.

Dado un punto M^i y la pose actual X_k del robot, la proyección en la imagen de la cámara n , $y_k^{i,n}$, se obtendrá a partir del modelo de proyección h :

$$y_k^{i,n} = h(t(M^i, X_k), P^n) \quad (3.10)$$

Para un robot y la cámara n , todas las medidas en un mismo instante de tiempo $\{y_k^{1,n}, \dots, y_k^{N,n}\}$ se agruparán en un vector de medidas Y_k^n .

$$Y_k^n = \begin{pmatrix} y_k^{1,n} \\ \vdots \\ y_k^{N,n} \end{pmatrix} \rightarrow Y_k = \begin{pmatrix} Y_k^1 \\ \vdots \\ Y_k^{N_c} \end{pmatrix}, \quad (3.11)$$

donde a su vez, el conjunto completo de medidas provenientes de todas las cámaras disponibles se encapsula en el vector Y_k .

Se define como $Y_k = h^a(X_k^a, P)$ a la ampliación de la ecuación de proyección, que relaciona el vector completo Y_k con el vector aumentado X_k^a y el conjunto de parámetros de la cámara P , para el caso general de N_c cámaras:

$$Y_k = h^a(X_k^a, P) = \begin{pmatrix} \left. \begin{matrix} h(t(X_k, M^1), P^1) \\ \vdots \\ h(t(X_k, M^N), P^1) \end{matrix} \right\} \text{Cámara1} \\ \vdots \\ \left. \begin{matrix} h(t(X_k, M^1), P^{N_c}) \\ \vdots \\ h(t(X_k, M^N), P^{N_c}) \end{matrix} \right\} \text{Cámara } N_c \end{pmatrix}. \quad (3.12)$$

Si se cuenta con una única cámara, entonces $Y_k = Y_k^1$ y $P = P^1$. La ordenación del vector Y_k y el número de elementos que lo componen puede en realidad sufrir modificaciones que serán en todo caso explícitamente comentadas en el texto. La situación más usual es que no todos los puntos del modelo del robot sean visibles en el plano imagen de las cámaras, y por tanto el vector Y_k no contendrá exactamente la proyección de todos ellos. En dichos casos, la relación que determina la asociación entre Y_k y los puntos $M^1, \dots, M^N$ será descrita específicamente.

3.4. Descripción de la pose y la proyección en la imagen como procesos aleatorios

En esta sección se aborda el modelado matemático de los procesos involucrados y anteriormente comentados desde un enfoque probabilístico. Existe en la literatura científica actual, relacionada con la robótica móvil, una tendencia clara de abordar el problema de la localización como uno descrito en términos de dinámica de procesos aleatorios. En los últimos años se han incorporado al ámbito de la localización, numerosas técnicas que hasta el momento pertenecían al terreno de la estadística matemática.

El enfoque propuesto es la interpretación de todos los procesos involucrados en la localización, es decir el movimiento del robot, las medidas que se realizan en la imagen, etc., como procesos aleatorios. Dicho cambio de perspectiva permite trasladar el problema de estimación de una magnitud, analíticamente relacionada con una serie de medidas indirectas de la misma, en un problema de inferencia Bayesiana.

Las técnicas de inferencia Bayesiana ofrecen un entorno compacto y sencillo en el cual formular preguntas acerca de parámetros a estimar, como puede ser la pose del robot, en función de toda la información que se precisa. La principal justificación, por la cual este cambio de enfoque tiene sentido, radica en el hecho que en todo proceso de estimación se tienen unos márgenes de incertidumbre debidos a ruido y fallos en modelos matemáticos imprecisos. Estos factores permiten casi de manera general asegurar que todo proceso en teoría determinista pero expuesto a procesos que tienen lugar en el mundo real, tiene asociado un proceso aleatorio que describe su incertidumbre. De esta manera si el objetivo de la tesis es encontrar la pose de un robot móvil, el proceso aleatorio asociado a la pose mostrará una función de densidad de probabilidad que define su comportamiento estadístico y que en términos de media coincide con el valor esperado del equivalente determinista.

Una clara ventaja de este enfoque es que permite relacionar la incertidumbre conocida

de determinados procesos, como puede ser la precisión con la que se es capaz de medir una determinada magnitud, con la repercusión directa de ésta en la solución buscada. Este hecho, al cual se referirá como capacidad de propagación de incertidumbre, permite aportar una medida muy útil de la precisión que se obtiene finalmente en la magnitud que es objetivo.

Bajo este mismo enfoque y razonamiento, se han desarrollado en la literatura técnicas muy eficaces relacionadas con el problema que se trata en la presente tesis, como puede ser el paradigma de la Localización Simultánea a la Reconstrucción de Entornos con robots móviles (SLAM).

En esta sección se justificará el modelado aleatorio de todos los procesos expuestos en este mismo capítulo.

En la Tabla 3.2 se establece la notación usada en la tesis referente a procesos aleatorios.

Los procesos aleatorios se definen en tipografía negrita (p. ej. $\mathbf{X}_k$ representa un proceso aleatorio y X_k una realización del mismo). De forma equivalente todo proceso aleatorio $\mathbf{X}_k$, se describe en el instante k mediante su función de densidad de probabilidad (f.d.p) $p(X_k)$. ▪ $p(X_k) = N(\hat{X}_k, \Sigma_k) \rightarrow$ representa una distribución Gausiana con media $\hat{X}_k$ y matriz de covarianza Σ_k. ▪ $\mathbf{X}_k \rightarrow$ proceso aleatorio que describe la pose del robot. ▪ $\mathbf{X}_k^a \rightarrow$ proceso aleatorio conjunto que representa la pose del robot y su estructura. ▪ $p(X_k X_{k-1}) \rightarrow$ función de densidad de probabilidad de tránsito para el proceso $\mathbf{X}_k$. ▪ $\mathbf{Y}_k \rightarrow$ proceso aleatorio que describe el vector de medidas Y_k. ▪ $p(Y_k X_k^a) \rightarrow$ función de verosimilitud de Y_k dado X_k^a. ▪ $p(X_k^a Y_1, \dots, Y_k) \rightarrow$ distribución de densidad de probabilidad “a posteriori” de X_k^a dado el conjunto de medidas en el intervalo $1, \dots, k$.

Tabla 3.2: Notación matemática y símbolos utilizados para describir los procesos aleatorios

3.4.1. Descripción de la pose del robot como un proceso aleatorio

La pose del robot, descrita mediante el vector X_k , se modela como un proceso aleatorio de Markov $\mathbf{X}_k$ con una función de densidad de probabilidad (f.d.p.) inicial $p(X_0)$ y una f.d.p. de transito $p(X_k|X_{k-1})$. Dicha función de tránsito se obtiene a partir de la propagación del proceso X_{k-1} a través del modelo de movimiento g e integrando en función de la incertidumbre asociada al vector U_k de odometría.

El hecho de modelar X_k como un modelo de Markov es consecuencia directa del carácter recursivo de la estimación. Es decir, la pose del robot en el instante k requiere el conocimiento de aquella en $k - 1$ y el vector de odometría U_k . Se supone que el vector

U_k de odometría está contaminado con ruido Gaussiano de media nula $\mathbf{W}_k = N(0, \Sigma_W)$ incorrelado con el proceso $\mathbf{X}_{k-1}$.

La ecuación en diferencias que define el proceso $\mathbf{X}_k$ es la siguiente:

$$\mathbf{X}_k = g(\mathbf{X}_{k-1}, U_k + \mathbf{W}_k), \quad (3.13)$$

donde los vectores que aparecen en negrita representan procesos aleatorios. La f.d.p de tránsito $p(X_k|X_{k-1})$ resulta de la solución de una ecuación integral de probabilidad en el que se elimina la influencia de la variable W_k :

$$p(X_k|X_{k-1}) = \int_{W_k} p(X_k|X_{k-1}, W_k)p(W_k)dW_k \quad (3.14)$$

Mediante la ecuación de tránsito $p(X_k|X_{k-1})$ se obtiene una regla recursiva para obtener el valor de $p(X_k)$.

$$p(X_k) = \int_{X_{k-1}} p(X_k|X_{k-1})p(X_{k-1})dX_{k-1} \quad (3.15)$$

La solución analítica de ecuaciones como la planteada en (3.15) resulta en general imposible, siendo necesario recurrir a métodos aproximados para obtener $p(X_k)$. Existen actualmente diversas alternativas en la literatura que permiten aproximar la expresión (3.15) y dar una solución suficientemente precisa. Dichas alternativas se pueden dividir en dos enfoques diferenciados:

- Descripción de variables aleatorias mediante métodos de Monte Carlo:

En dichos métodos, las distribuciones de probabilidad se aproximan por un conjunto de muestras cuyas propiedades estadísticas muestrales coinciden con aquellas de la f.d.p. que se busca. Mediante dichas muestras, la propagación de estadísticos a través de funciones no lineales es sencillo. Su tiempo de cómputo es, sin embargo, elevado, y el número de muestras crece exponencialmente con las dimensiones de la variable aleatoria aproximada. Dentro de este tipo de técnicas se engloban los ampliamente utilizados filtros de partículas.

- Descripción de variables aleatorias mediante aproximaciones funcionales:

En este caso, las funciones de densidad de probabilidad son aproximadas por una f.d.p. definida analíticamente (p. ej. funciones Gaussianas). Su tránsito por funciones no lineales se realiza mediante métodos de aproximación de Taylor o similares.

El segundo tipo de método, utilizando funciones de densidad de probabilidad gaussianas, es la alternativa más utilizada en procesos de similar complejidad al propuesto, como puede ser el problema SLAM [Davison, 2003] utilizando cámaras. En este tipo de alternativa, por cada proceso, tan solo la media y la varianza serían suficientes para su descripción completa.

La aproximación gaussiana, no obstante, se debe efectuar cada vez que un proceso es propagado a través de funciones no lineales (p. ej. $g(X_{k-1}, U_k)$), ya que el resultado no es en general una f.d.p. Gaussiana.

La utilización de funciones gaussianas para describir procesos aleatorios está ampliamente documentada en la literatura [Van der Merwe et al., 2000], existiendo numerosas alternativas que aseguran consistencia en las aproximaciones, como puede ser la transformada Unscented. En la presente tesis se utilizarán aproximaciones de primer orden de las funciones no lineales para la propagación de variables aleatorias Gaussianas.

Acudiendo a los métodos de propagación anteriormente comentados, la aproximación Gaussiana de la pose $\mathbf{X}_k$, con f.d.p. $N(\hat{X}_k, \Sigma_k)$, admite una descripción recursiva sencilla utilizando una aproximación de primer orden de g :

$$p(X_k) = N(\hat{X}_k, \Sigma_k) \quad \begin{cases} \hat{X}_k = g(\hat{X}_{k-1}, U_k) \\ \Sigma_k = J_{gx} \Sigma_{k-1} J_{gx}^T + J_{gu} \Sigma_W J_{gu}^T \end{cases} \quad (3.16)$$

donde J_{gx} y J_{gu} son los jacobianos de la función g en función de X_{k-1} y U_k respectivamente. La regla recursiva requiere además definir $p(X_0) = N(\hat{X}_0, \Sigma_0)$, como la aproximación Gaussiana de la pose inicial. En la Figura 3.2, se observa el comportamiento del equivalente Gaussiano obtenido del proceso, comparado con una simulación de Monte Carlo del mismo.

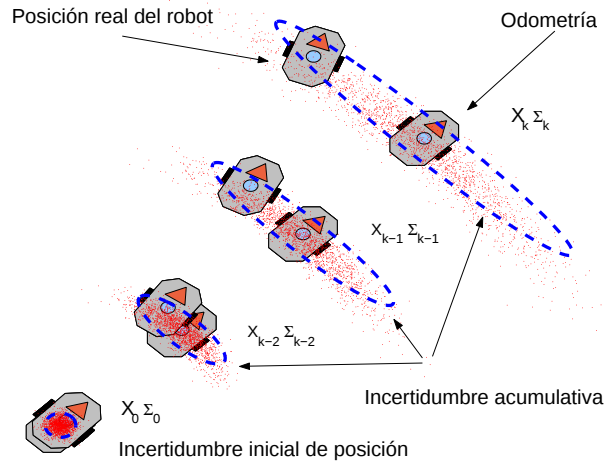


Figura 3.2: Visualización del proceso $\mathbf{X}_k$ a través del tiempo mediante la utilización de elipses de incertidumbre que describen la covarianza del proceso en sus dos primeras componentes (posición en el plano). Se representa cada muestra de una simulación de Monte Carlo como un punto que representa la posición del robot en el plano. Como puede contemplarse, la incertidumbre de la distribución real, representada por el conjunto de puntos de Monte Carlo, se dispersa a medida que el robot se mueve. Dicho esparsamiento es predicho y modelado mediante las elipses de incertidumbre del proceso Gaussiano.

3.4.2. Proceso aleatorio conjunto de pose y geometría

En este apartado se comenta brevemente el modelado del vector X_k^a como proceso aleatorio $\mathbf{X}_k^a$. El principal motivo de centrar la atención en dicho vector proviene del hecho de que la relación estadística entre la pose X_k y los puntos M demostrara ser de importancia para conseguir una estimación estable de la pose.

El método utilizado para describir X_k^a como un proceso aleatorio opera de forma equivalente al utilizado para describir la pose. En este caso se describirá directamente suponiendo que todos los procesos son Gaussianos. El proceso $\mathbf{X}_k^a$ se modela como un proceso aleatorio de Markov con f.d.p. inicial $p(X_0^a) = N(\hat{X}_0^a, \Sigma_0^a)$ y función de transición de estados $p(X_k^a | X_{k-1}^a)$. La transición de estados es la consecuencia directa de aplicar el modelo de movimiento g^a a los procesos aleatorios de entrada. Al igual que se comentó con anterioridad, el vector de entrada U_k se supone contaminado con el ruido $\mathbf{W}_k$.

Para cada instante de tiempo k la solución para la pose y la geometría del robot se aproxima como un proceso Gaussiano $\mathbf{X}_k^a$, con media $\hat{X}_k^a$ y matriz de covarianza Σ_k^a .

$$\hat{X}_k^a = \begin{pmatrix} \hat{X}_k \\ \hat{M}^1 \\ \vdots \\ \hat{M}^N \end{pmatrix} \quad \Sigma_k^a = \begin{pmatrix} \Sigma_X & \Sigma_{XM^1} & \cdots & \Sigma_{XM^N} \\ \Sigma_{M^1X} & \Sigma_{M^1} & \cdots & \Sigma_{M^1M^N} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{M^NX} & \Sigma_{M^NM^1} & \cdots & \Sigma_{M^N} \end{pmatrix} \quad (3.17)$$

3.4.3. Proceso de medida en la imagen como proceso aleatorio

El proceso de medida, el cual da lugar al vector Y_k mediante una o varias cámaras, posee una caracterización como proceso aleatorio $\mathbf{Y}_k$, cuyas propiedades se obtienen acudiendo a la función de proyección h^a :

$$\mathbf{Y}_k = h^a(\mathbf{X}_k^a, \mathbf{P}) + \mathbf{V}_k, \quad (3.18)$$

donde $\mathbf{P} = N(\hat{P}, \Sigma_P)$ representa un proceso Gaussiano que determina la incertidumbre en el vector $\mathbf{P}$ (que engloba los parámetros de calibración del conjunto de cámaras). El proceso $\mathbf{V}_k = N(0, \Sigma_V)$ representa el error de detección cometido en Y_k .

A continuación se detalla la estructura y propiedades de ambos procesos y fuentes de incertidumbre en el vector Y_k .

Incetidumbre en la calibración de cámaras

Los algoritmos de calibración, utilizados para calibrar el conjunto de cámaras que forman el “Espacio Inteligente”, no son exactos e introducen un error o incertidumbre en el vector de parámetros P . Las propiedades de dicha incertidumbre de calibración varían con el método utilizado y las condiciones en las que se ha realizado. En general, el proceso equivalente $\mathbf{P}$ se define mediante un vector media $\hat{P}$ que coincide con el resultado numérico

de la calibración, y una matriz de covarianza Σ_P , que modela el error cometido en los parámetros P .

En general, la estructura de la matriz de covarianza Σ_P varía según las propiedades estadísticas de la incertidumbre en los parámetros de calibración y según el método de calibración utilizado. En la presente tesis supondremos por simplicidad que en todo caso la correlación estadística entre sus elementos es nula y por tanto Σ_P corresponde con una matriz diagonal:

$$\Sigma_P = \begin{pmatrix} \Sigma_{P^1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Sigma_{P^{N_c}} \end{pmatrix} \quad \Sigma_{P^n} = \begin{pmatrix} \sigma_{\alpha_c^n}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{\beta_c^n}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\gamma_c^n}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Sigma_{T_c^n} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{f_u^n}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{f_v^n}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{u_0^n}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{v_0^n}^2 \end{pmatrix} \quad (3.19)$$

Los valores utilizados en su diagonal son obtenidos fácilmente a partir de un método de calibración basado en patrones de calibración, como puede ser el método de Tsai [Tsai, 1986], el cual es utilizado en esta tesis como técnica de calibración del conjunto de cámaras.

Incetidumbre en la detección

Las componentes que forman el vector Y_k corresponden a la proyección de la estructura del robot en la imagen o imágenes de las cámaras. Cada medida se obtiene mediante métodos de detección de marcas naturales, los cuales son comentados en el capítulo 4.

Todo proceso de medida en la imagen está sujeto a varias fuentes de error que afectan al vector Y_k y que permiten representarlo como un proceso aleatorio $\mathbf{Y}_k$. En primer lugar, los métodos de detección presentan un margen de error en la localización de un mismo punto del robot en diferentes posiciones y orientaciones, lo que se traduce en incertidumbre en Y_k . En añadido a dicha fuente de error, el plano imagen sufre de discretización espacial y cuantificación en la intensidad de luz adquirida por el sensor.

La incertidumbre cometida en la detección se considera independiente, tanto entre las diferentes cámaras, como en las medidas puntuales $y_k^{i,n}$ incluidas en Y_k . Se modela mediante un proceso $\mathbf{V}_k = N(0, \Sigma_{V_k})$ de ruido aditivo de media nula que incrementa la incertidumbre del proceso $\mathbf{Y}_k$ en los mismos términos que los fenómenos reales comentados. La matriz de covarianza Σ_V es una matriz diagonal por bloques:

$$\Sigma_{V_k} = \begin{pmatrix} \Sigma_{V_k}^{1,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Sigma_{V_k}^{N,N_c} \end{pmatrix}, \quad (3.20)$$

donde cada bloque $\Sigma_{V_k}^{i,n}$, consiste en una matriz 2×2 que representa la incertidumbre en la medida de un solo punto en la imagen. En general, cada bloque se representa a su vez por una matriz diagonal con un valor σ_v^2 en su diagonal principal, cuyo valor es dependiente del método de detección utilizado.

El proceso $\mathbf{Y}_k$, se define por tanto a través su media estadística $\hat{Y}_k$ y covarianza Σ_{Y_k} , resultado de propagar los procesos de los que depende a través de la expresión (3.18):

$$\hat{Y}_k = h^a(\hat{X}_k^a, \hat{P}) \quad \Sigma_{Y_k} = J_{hx} \Sigma_k^a J_{hx}^T + J_{hP} \Sigma_P J_{hP}^T + \Sigma_V, \quad (3.21)$$

donde J_{hX} y J_{hP} corresponden con los Jacobianos de la función h^a con respecto a X_k^a y P respectivamente.

Como se hace patente en la expresión (3.21), existen diferentes fuentes de incertidumbre que conforman el proceso $\mathbf{Y}_k$ y entre las cuales se ha incluido el proceso $\mathbf{X}_k^a$. Sin embargo, cabe destacar que en general el hecho de incluir la incertidumbre del vector X_k^a tiene sentido tan solo cuando se observa el proceso Y_k como una variable aleatoria estimada y se propaga la incertidumbre de X_k^a al plano imagen.

En el caso más intuitivo de que Y_k se trate como un proceso de detección en la imagen de un punto tridimensional determinista, el proceso de medida tan solo se ve afectado por los errores de calibración y detección:

$$\mathbf{Y}_k = h^a(X_k^a, \mathbf{P}) + \mathbf{V}_k, \quad (3.22)$$

dando lugar a la siguiente distribución Gausiana:

$$\hat{Y}_k = h^a(X_k^a, \hat{P}) \quad \Sigma_{Y_k} = J_{hP} \Sigma_P J_{hP}^T + \Sigma_V. \quad (3.23)$$

La matriz de covarianza Σ_{Y_k} de la expresión (3.23), posee una estructura particular, la cual refleja las dependencias estadísticas de los elementos del vector Y_k . Como se pondrá de manifiesto en esta tesis, los elementos de correlación cruzada, que establecen dichas dependencias, juegan un papel importante en aspectos del algoritmo de localización propuesto, como puede ser el tiempo de cómputo o la precisión obtenida.

Debido al hecho de que todas las medidas de una misma cámara comparten los mismos parámetros de calibración, y dada la estructura diagonal de Σ_P la matriz Σ_{Y_k} será diagonal por bloques. Cada bloque engloba a todas las medidas tomadas con la misma cámara:

$$\Sigma_{Y_k} = \begin{pmatrix} \Sigma_{Y_k^1} & \cdots & \bar{0} \\ \vdots & \ddots & \vdots \\ \bar{0} & \cdots & \Sigma_{Y_k^{N_c}} \end{pmatrix} \quad (3.24)$$

Para una sola cámara, la estructura de la matriz Σ_{Y_k} , la cual corresponden con la de uno sólo de los bloques (p. ej. $\Sigma_{Y_k^1}$) de la expresión (3.24), se observa a continuación:

$$\begin{aligned}
 \Sigma_{Y_k} = & \underbrace{\left(\begin{array}{c} \text{Matrix } J_{hP} \end{array} \right)}_{J_{hP}} \underbrace{\left(\begin{array}{c} \text{Matrix } \Sigma_P \end{array} \right)}_{\Sigma_P} \underbrace{\left(\begin{array}{c} \text{Matrix } J_{hP}^T \end{array} \right)}_{J_{hP}^T} + \underbrace{\left(\begin{array}{c} \text{Matrix } \Sigma_{V_k} \end{array} \right)}_{\Sigma_{V_k}} \quad (3.25) \\
 & \underbrace{\hspace{15em}}_{\Sigma_{Y_k} = \left(\begin{array}{c} \text{Matrix } \Sigma_{Y_k} \end{array} \right)}
 \end{aligned}$$

En la práctica, y como se demostrará en posteriores capítulos, en general el modelado completo de la matriz Σ_{Y_k} se podrá simplificar por una aproximación diagonal, la cual es suficiente para obtener un sistema de localización preciso.

3.5. Conclusiones

El presente capítulo ha presentado la notación y modelos matemáticos necesarios para proponer un sistema de localización de robots mediante las cámaras del “Espacio Inteligente”. Los modelos utilizados, tanto de dinámica de movimiento como aquellos que describen la proyección en las cámaras, son presentados como procesos aleatorios, en los que la incertidumbre interviene directamente en el problema. El objetivo es describir la localización del robot como la un proceso de estimación de variables aleatorias Gaussianas. Las distribuciones Gaussianas modelan la incertidumbre mediante la covarianza y el valor numérico de la estimación mediante su media. De este modo, la aproximación mediante variables aleatorias Gaussianas es altamente descriptiva cuando es aplicable.

En los siguientes capítulos, se procederá a la utilización de los modelos presentados, incluyendo su descripción estadística, para la consecución de los objetivos planteados.

Capítulo 4

Detección de marcas naturales y procesos de seguimiento

4.1. Introducción

El objetivo del proceso de medida, necesario para la estimación de la pose del robot, es la obtención de las coordenadas en el plano imagen, de puntos pertenecientes a su estructura tridimensional. En la presente tesis, el proceso de medida en el plano imagen es utilizado como la fuente de información fundamental que permite relacionar la pose del robot con las imágenes tomadas por el conjunto de cámaras. En este capítulo se plantean soluciones específicas para las diferentes problemáticas que se abordan en la tesis. Para ello se recurre a técnicas de detección y correspondencia de marcas naturales, ampliamente extendidas en visión artificial, y que previamente fueron introducidas en el capítulo 2. Las principales contribuciones de este capítulo consisten en el diseño a medida realizado para cada situación específica que se desarrolla en el planteamiento de la tesis, poniendo de manifiesto la necesidad de acudir a la sinergia entre diferentes técnicas para dar lugar a una solución satisfactoria para el proceso de observación.

El proceso de medida se divide de manera fundamental en dos tareas:

- **Proceso de detección de marcas naturales:** en este proceso se obtienen aquellas coordenadas de la imagen que son candidatos a ser identificados como puntos de la estructura tridimensional del robot.
- **Mecanismo de correspondencia:** a partir del conjunto de puntos detectados en una imagen, el mecanismo de correspondencia permite identificar aquellos que componen la estructura del robot. La identificación puede ser en base a similitudes en la apariencia local de la imagen o cercanía geométrica.

Aunque ambos procesos están esencialmente relacionados entre sí, existe, como veremos en este capítulo, la posibilidad de proponer diferentes sistemas de detección para un mismo tipo de proceso de correspondencia.

A continuación se presenta el problema fundamental y los requisitos exigidos en el proceso de medida.

4.1.1. Situaciones planteadas en la tesis

Dada la estructura del algoritmo propuesto, existen básicamente tres escenarios de medida, los cuales difieren esencialmente en la cantidad y naturaleza de información “a priori” que se tiene y que asimismo modifica las restricciones impuestas al proceso de observación.

A continuación se describen dichas situaciones y los requerimientos específicos de cada una con respecto al proceso de medida en la imagen. El orden elegido para enumerar cada uno de los escenarios será el mismo que el utilizado en el resto del documento:

- **Seguimiento de marcas durante la inicialización:** En la fase de inicialización, la cual se comenta específicamente en el capítulo 5, se desconoce la estructura del robot y su apariencia. De este modo se carece de un modelo previo del robot así como el

valor de su pose. Los mecanismos de inicialización, en especial aquellos que cuentan con una sola cámara, requieren la obtención de una serie de trayectorias o “tracks” visuales que se corresponden con puntos de la estructura del robot en movimiento.

Existen en esta situación dos problemas fundamentales relacionados con el proceso de medida:

1. Del conjunto de puntos detectados en una imagen, tan solo son válidos aquellos que pertenecen a la estructura del robot. Se debe encontrar un método eficiente que descarte puntos que pertenecen al fondo de la escena, utilizando para ello, análisis de movimiento o el conocimiento de una imagen del fondo utilizada como patrón.
2. Durante la inicialización se desconoce la pose y geometría del robot, lo que consecuentemente impide el uso de la odometría como estimación de movimiento. De este modo, no existe conocimiento preciso de la posición que cada punto detectado ocupará en el siguiente instante de tiempo. La correspondencia de un mismo punto en diferentes instantes de tiempo se debe solucionar atendiendo únicamente a la información proveniente del plano imagen y, en menor medida, la suposición de movimiento reducido del robot entre cada instante de tiempo.

■ **Detección y correspondencia de marcas naturales en procesos de Estimación-**

Corrección: Esta situación corresponde con el sistema de obtención secuencial de la pose del robot, que se presenta en el capítulo 7. En dicho proceso se posee un alto grado de conocimiento acerca de la posición, proyectada en el plano imagen, de los puntos o marcas naturales utilizadas en cada instante de tiempo. En este caso, se conoce la estructura tridimensional del robot y la pose anterior que ocupa. Mediante esta información y la estimación de movimiento, la posición de las marcas en un nuevo instante de tiempo está acotada mediante una región de incertidumbre, dada por la descripción como procesos aleatorios de la pose y las medidas en el plano imagen del robot.

En este problema, dada la existencia de una cantidad de información elevada, se plantea un sistema de “matching” de marcas naturales basado en la teoría de la compatibilidad estadística.

■ **Detección y emparejamiento de marcas naturales en configuración “Wide-Baseline”**

En este último tipo de situación, el modelo tridimensional del robot es conocido, pero no la pose anterior. Este tipo de situación atiende al problema del secuestro o instantes de reinicialización del sistema, en los cuales se pierde el robot y el proceso secuencial no es válido. Esta situación requiere un sistema de detección y emparejamiento de una marca ante diferentes poses del objeto. A diferencia del proceso de inicialización, en este caso no se puede suponer un movimiento suave entre diferentes instantes de tiempo. Sin embargo, el conocimiento de un modelo tridimensional, permite aplicar un esquema robusto que será descrito en el capítulo 7 y por tanto es posible acudir a técnicas de emparejamiento “wide-baseline”. Dichas técnicas, a pesar de introducir una cantidad importante de “outliers” o falsos emparejamientos, son capaces de identificar el robot con independencia de la pose que éste ocupa.

El capítulo está estructurado por tanto en dos partes fundamentales: en primer lugar se tratará el problema de detección, proponiendo una solución única que sea esencialmente válida en todas las situaciones planteadas con anterioridad. En la segunda parte del capítulo se aborda el problema de correspondencia entre puntos detectados, el cual se divide en los tres escenarios diferentes, correspondientes a la inicialización, proceso de estimación-corrección y el problema del secuestro.

4.2. Detección de marcas naturales

El proceso de detección presentado en esta tesis tiene como objetivo la localización en la imagen de aquellos puntos de interés correspondientes a la estructura del robot.

La detección de marcas naturales debe cumplir con los siguientes requerimientos:

- **Estabilidad ante cambios de perspectiva:** un punto de interés debe ser persistente en la imagen ante cambios de la pose de la cámara con respecto al objeto que da lugar a la imagen. Dichos cambios incluyen cambios de escala, transformaciones proyectivas y en general son dependientes de la estructura tridimensional que da lugar al punto buscado.
- **Localidad espacial:** las marcas naturales detectadas, las cuales se representarán por sus coordenadas en la imagen, deben corresponder a un punto bien localizado en el espacio tridimensional.
- **Tiempo de cómputo asumible:** El método utilizado para su detección en la imagen debe implicar una complejidad algorítmica admisible por un sistema de procesamiento actual.

En el capítulo 2 se han comentado brevemente los métodos de detección y seguimiento de marcas naturales en configuraciones “wide-baseline”. Dichos métodos, además de proponer estructuras de identificación invariantes a transformaciones afines, como puede ser el descriptor SIFT [Lowe, 1999], SURF [Bay et al., 2006] y similares, dan solución al problema de la detección mediante métodos de detección multi-escala.

Los métodos de detección multi-escala se basan, en su gran mayoría, en un análisis de la imagen en diferentes escalas de la misma. El espacio multi-escala se construye de manera general mediante filtros gaussianos isotrópicos con diferentes valores para la varianza σ en función de la escala deseada. Si $I(u, v)$ es la imagen original, su representación multi-escala se representa por $L(u, v, \sigma)$.

A partir de dicho espacio de escala se diseña un operador, que permite encontrar, normalmente mediante búsqueda de extremos locales, regiones de interés definidas por sus coordenadas espaciales y escala. Esta idea fue propuesta por primera vez por [Crowley, 1982] y [Crowley and Parker, 1984], utilizando una estructura piramidal de diferencias de gaussianas en cuyos extremos locales se encuentran las localizaciones de los puntos de interés.

Los métodos existentes de detección, entre los que se encuentra el método DoG propuesto por [Lowe, 1999] y utilizado en esta tesis, se diferencian en la definición y propiedades del espacio de escala que utilizan, lo que da como resultado diferentes estructuras detectadas en la imagen.

Se pueden distinguir dos metodologías diferentes:

- Detectores de esquinas multi-escala mediante la matriz de estructura: en este tipo de métodos se amplía el concepto de matriz de estructura, útil en la detección de esquinas en la imagen (método “Harris”), para su uso en un espacio multiresolución. Este tipo de método se conoce en la literatura como Harris-Multiescala, y ha sido propuesto como base para diferentes métodos de detección, incluyendo un método con invarianza afín completa.
- Detectores de regiones o “blobs” en un espacio de escala: en este tipo de métodos, las regiones obtenidas corresponden a puntos extremos de un espacio multiresolución, resultado de aplicar filtros “Laplaciana de Gaussiana”, o en su defecto filtros de respuesta parecida, como las diferencias de gaussianas “Difference of Gaussians” (DoG). Los máximos de dicho espacio permiten encontrar regiones denominadas “blobs”, definidas en posición y escala. Los denominados “blobs” corresponden con áreas con textura, vecindad de esquinas y puntos cercanos a bordes en la imagen. Dentro de este tipo de métodos se encuentra clasificado el sistema de detección incluido en el método SIFT propuesto por [Lowe, 1999], comentado previamente en el capítulo 2.

En general, los detectores de blobs, como el método SIFT, y aquellos explícitamente sensibles a esquinas, como el detector “Harris-Laplace”, anteriormente comentados, no producen el mismo tipo de características. Como se ha comentado con anterioridad, el método SIFT es sensible a los denominados “blobs”, que consisten en regiones generalmente circulares, que poseen contraste con la textura adyacente. En imágenes naturales, los “blobs” aparecen con mucha frecuencia, incluyendo vecindades de bordes y de esquinas. El método SIFT incluye una etapa de filtrado se eliminan aquellos “blobs” considerados poco estables, como pueden ser las vecindades de bordes.

Este tipo de análisis produce en la práctica que el método SIFT detecte esquinas, aunque en realidad el máximo del espacio *DoG* se produce en una vecindad de la misma.

El método “Harris-Multiescala” es sensible principalmente a esquinas, donde la matriz de estructura ofrece un máximo. En este método, otras estructuras, como ciertos tipos de “blobs” o zonas de alto contenido en textura, pueden ser incluidas como puntos detectados.

El resultado neto es que ambos métodos proporcionan características muy cercanas y son comparables en términos de estabilidad. En esta tesis se propone el uso del método propuesto por Lowe, basado en una estructura piramidal de diferencias de gaussianas (DoG), el cual es detallado en [Lowe, 1999].

A continuación se resume el detector utilizado en el método SIFT como referencia en el texto.

4.2.1. Detectores de regiones o “blobs” en un espacio de escala

En [Lindeberg, 1994], se desarrolla un espacio de escala definido mediante la aplicación sucesiva del operador Laplaciana de Gaussiana (LoG) a la imagen:

Dada una función gaussiana de convolución de varianza σ^2 ,

$$g(u, v, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(u^2+v^2)/(2\sigma^2)}, \quad (4.1)$$

se construye el espacio de escala $L(u, v, \sigma)$ de la imagen $I(u, v)$ mediante la convolución del operador $g(u, v, \sigma)$:

$$L(u, v, \sigma) = g(u, v, \sigma) * I(u, v), \quad (4.2)$$

el espacio $LoG(u, v, \sigma)$, correspondiente a la imagen $I(u, v)$ será por tanto:

$$LoG(u, v, \sigma) = \sigma^2 |\nabla^2 L(u, v, \sigma)| = \sigma^2 |L_{uu} + L_{vv}| \quad (4.3)$$

En [Lindeberg, 1994] se muestra que los extremos del espacio $LoG(u, v, \sigma)$ determinan localizaciones en escala y posición en la imagen muy estables para su uso como detectores de marcas naturales:

$$(u, v, \sigma) = \operatorname{argmaxmin}_{(u,v,\sigma)} (LoG(u, v, \sigma)). \quad (4.4)$$

Cada área de interés, extremo de la función $LoG(u, v, \sigma)$ se define por su posición u, v y la escala correspondiente, determinada por σ . El operador LoG tiene la propiedad de asegurar localidad tanto en las posiciones como por la escala. De esta manera, una región (u_0, v_0, σ_0) detectada en $I(u, v)$, dará lugar al punto $(su_0, sv_0, s\sigma_0)$ en la versión escalada $I(su, sv)$.

El método SIFT [Lowe, 1999] aproxima el espacio $LoG(u, v, \sigma)$ mediante un espacio piramidal de diferencias de Gaussianas $DoG(u, v, \sigma)$:

$$LoG(u, v, \sigma) \approx DoG(u, v, \sigma) = L(u, v, \sigma) - L(u, v, s\sigma), \quad s < 1 \quad (4.5)$$

En el método SIFT se dividen las posibles escalas aplicadas al parámetro σ en octavas, aplicando un escalado a la imagen en cada octava, lo que permite reducir la carga computacional:

$$\sigma(o, s) = \sigma_0 2^{o + \frac{s}{S}}, o \in \operatorname{omin} + [0, \dots, O - 1], s \in [0, \dots, S - 1] \quad (4.6)$$

Los extremos de $LoG(u, v, \sigma)$ o en su defecto el espacio $DoG(u, v, \sigma)$, corresponden con “blobs”, cuyas propiedades han sido comentadas anteriormente. Los “blobs”, en general, consisten en regiones circulares, esquinas o bordes. Los bordes y regiones de poco contraste dan lugar a características muy poco estables que son eliminadas en el método SIFT mediante el análisis de las derivadas del espacio $DoG(u, v, \sigma)$ en torno al punto detectado

en cuestión. Además, Lowe propone un método que permite refinar de forma subpíxelica tanto la escala como la posición de cada punto de interés, utilizando un algoritmo iterativo de tipo Gauss-Newton, lo cual incrementa la estabilidad del problema.

El método SIFT asigna además, una orientación principal θ a cada punto de interés detectado, con el objetivo de corregir dicha orientación en cada detección.

En resumen, el método de detección propuesto por Lowe identifica cada punto de interés mediante su posición, orientación y escala.

4.3. Mecanismo de correspondencia de marcas naturales

En esta sección se aborda el problema de la correspondencia de puntos de interés en imágenes de un objeto ante transformaciones de perspectiva, particularizada a los tres escenarios principales anteriormente indicados.

De manera general, como se ha comentado en el capítulo 2, existen numerosos estudios que dan solución al problema de correspondencia, esto es, encontrar un mismo punto de la geometría de un objeto ante cambios de perspectiva, utilizando para ello su apariencia local en el plano imagen.

A partir de dicha información de apariencia, se establecen descripciones compactas o descriptores, sobre los cuales recae la identificación de un mismo punto en diferentes imágenes.

En general, se buscan descripciones invariantes a transformaciones afines, como el método SIFT propuesto por [Lowe, 1999] o el método SURF, derivado de este último y propuesto por [Bay et al., 2006]. Todos estos métodos suponen que la deformación, producida por la imagen en torno a un punto de interés, se modela mediante una transformación afín. Por tanto, la invarianza ante dicha transformación asegura un método suficientemente robusto para ser utilizado en el emparejamiento de puntos ante cambios de perspectiva.

En este capítulo se pone de manifiesto la dificultad existente en la aplicación directa de ese tipo de métodos en cualquiera de los escenarios de correspondencia propuestos en la tesis.

En la figura 4.1 se muestra un ejemplo ilustrativo de la clase de imágenes obtenidas del robot mediante una cámara del “Espacio Inteligente”. En dicha figura se han separado cuatro instantes de tiempo de una misma trayectoria. En cada instante se han marcado una serie de puntos de interés de la estructura del robot, obtenidos mediante el método de detección DoG, propuesto por Lowe.

Como se puede ver en la figura 4.2, la información de apariencia, local a cada punto, es representada aumentada para los diferentes puntos de interés e instantes de tiempo. Las diferencias entre los distintos instantes de tiempo muestran que la textura local a cada punto no se modela mediante una transformación afín, puesto que incluye el fondo de la escena variable y cambios debidos a una estructura local no plana.

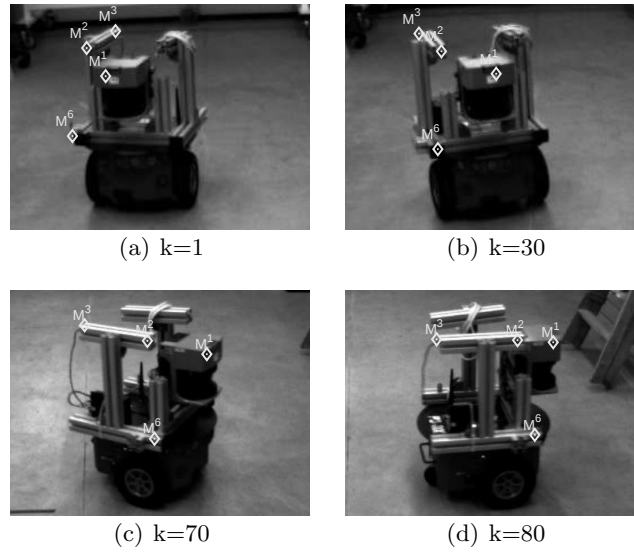


Figura 4.1: Instantáneas de una trayectoria de un robot visto por una cámara.

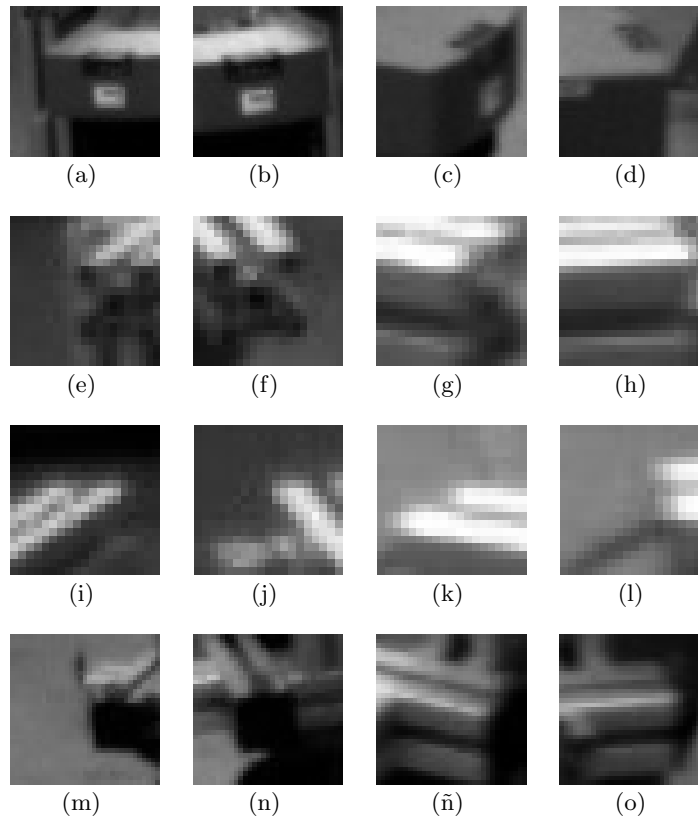


Figura 4.2: Imagen de la textura local de los puntos de interés detectados en diferentes instantes de tiempo. Cada fila corresponde con un mismo punto detectado en los instantes de tiempo $k = 1, 30, 70, 80$, mostrados en la figura 4.1.

Esta situación se repite con mucha frecuencia en estructuras artificiales, como los robots que se pretende localizar, y por tanto dificultan enormemente el uso convencional de los algoritmos disponibles en la literatura. De este modo, muchas de las decisiones y propuestas de este capítulo son debidas a la aplicación a objetos tridimensionales, que además presentan un reducido tamaño en el plano imagen y dan lugar a puntos de interés con un alto grado de variabilidad en su textura local.

En la tabla 4.1, se muestra la distancia euclídea existente entre los descriptores SIFT asociados entre los diferentes instantes de tiempo mostrados en la figura 4.2. Cada uno de los cuadros incluidos en la tabla 4.1 compara los descriptores encontrados en el instante $k = 1$ de la secuencia $d_1^1, \dots, d_1^4$ con aquellos encontrados en los instantes siguientes.

	d_1^1	d_1^2	d_1^3	d_1^4		d_{30}^1	d_{30}^2	d_{30}^3	d_{30}^4
d_1^1	0,000000	1,104635	0,935032	1,181701	d_{30}^1	0,736385	0,973158	1,001406	1,063666
d_1^2	1,104635	0,000000	1,140407	0,915369	d_{30}^2	0,989539	0,690644	1,057129	1,083076
d_1^3	0,935032	1,140407	0,000000	1,176118	d_{30}^3	0,918698	1,237011	0,652730	1,198232
d_1^4	1,181701	0,915369	1,176118	0,000000	d_{30}^4	1,107769	1,032919	1,102268	1,028279
(a)					(b)				
	d_{70}^1	d_{70}^2	d_{70}^3	d_{70}^4		d_{80}^1	d_{80}^2	d_{80}^3	d_{80}^4
d_1^1	0,679031	0,941885	0,946884	1,099332	d_{80}^1	1,077291	1,106085	1,029785	1,118644
d_1^2	1,091547	0,630873	1,176236	0,987247	d_{80}^2	0,939313	1,259635	0,765161	1,298393
d_1^3	0,871057	1,021180	0,937862	1,098850	d_{80}^3	1,111963	1,060942	1,233001	1,131708
d_1^4	0,971100	1,150993	1,088985	1,138123	d_{80}^4	1,173220	1,069935	1,157590	1,000974
(c)					(d)				

Tabla 4.1: Tablas que muestran la distancia euclídea entre los descriptores de los puntos señalados en la figura 4.2 en los distintos instantes de tiempo. El valor indexado por cada posición de cada tabla indica la distancia entre los descriptores del primer instante de tiempo $k = 1$ ($d_1^1, \dots, d_1^4$), distribuidos en las filas, con los descriptores obtenidos en los diferentes instantes de tiempo $k = 1, 30, 70, 80$, mostrados en las columnas.

Los elementos de la diagonal principal de cada una de las tablas representan la distancia asociada al emparejamiento correcto, es decir, el “matching” entre un mismo punto en diferentes instantes de tiempo. Si el método SIFT ofrece un emparejamiento correcto, los elementos de la diagonal principal deberían contener un valor menor que cualquier otro perteneciente a su correspondiente fila. De otro modo, el emparejamiento es incorrecto. Como se puede ver con nitidez en los elementos sombreados de la tabla 4.1, los cuales marcan el mínimo de cada fila, la degradación sufrida por los cambios de perspectiva produce que el método SIFT produzca fallos en el emparejamiento de puntos a partir del instante $k = 30$. Este ejemplo, aunque sencillo, muestra de manera muy explícita la problemática existente en las condiciones impuestas en la tesis para la aplicabilidad de descriptores SIFT como solución al proceso propuesto.

A continuación se detalla el proceso de medida en los diferentes escenarios comentados en la tesis.

4.3.1. Correspondencia de puntos en la fase de inicialización del sistema

El objetivo de la fase de inicialización es la obtención de la primera porción de estructura y pose del robot, para su uso en un algoritmo online posterior, que consiste en un proceso de estimación-corrección de Kalman.

El método de correspondencia de puntos en la fase de inicialización se divide esencialmente en dos partes fundamentales:

- Eliminación de puntos de interés pertenecientes al fondo de la escena.
- Obtención de “trayectorias visuales” a partir de un conjunto de imágenes del robot en movimiento.

Eliminación de puntos pertenecientes al fondo

Puesto que las cámaras del “Espacio Inteligente” son fijas, el fondo puede ser considerado estático, y por tanto se supone la disponibilidad de una descripción de dicho fondo cuando no existe un robot en él. La eliminación de aquellos puntos detectados en la imagen que pertenecen a la escena estática es posible de una manera sencilla y robusta.

Se parte de una imagen de fondo, tomada en ausencia de robots u objetos en movimiento. A partir de dicha imagen se obtiene un conjunto de puntos $\mathcal{C}_f = \{y_f^1, \dots, y_f^{N_f}\}$, mediante cualesquiera de los métodos de detección comentados con anterioridad. Utilizando las posiciones $\mathcal{C}_f$, y añadiendo el valor de escala y orientación disponibles por el método de detección, se extrae el conjunto de descriptores SIFT $\mathcal{D}_f = \{d_f^1, \dots, d_f^{N_f}\}$. Ambos conjuntos $\mathcal{C}_f$ y $\mathcal{D}_f$ representan el modelo de fondo utilizado.

Se obtienen del mismo modo los conjuntos $\mathcal{C}_k$ y $\mathcal{D}_k$ detectados en una imagen del robot en movimiento. Se plantea un algoritmo de búsqueda de correspondencias entre el fondo y la escena actual, utilizando para ello los conjuntos $\mathcal{D}_f$ y $\mathcal{D}_k$.

Puesto que si un punto de la escena actual pertenece al fondo, su apariencia y posición en la imagen debe permanecer idealmente intacta, basta con encontrar aquellos puntos de la imagen actual que tienen como correspondiente SIFT uno de la imagen utilizada como fondo de la escena. La correspondencia SIFT se realiza mediante la distancia euclídea de los descriptores $|d_f^i - d_k^j|$ asociados a cada punto detectado en el fondo y la imagen respectivamente. Para incrementar la estabilidad de la segmentación, el emparejamiento es guiado, buscando las correspondencias de un punto concreto del fondo en la imagen actual dentro de un radio de vecindad. Es decir, sea un punto de la imagen actual y_k^i y su descriptor asociado d_k^i , entonces se considerará que pertenece al fondo si existe un punto y_f^j del fondo tal que $|d_f^j - d_k^i| < d_{max}$ y $|y_f^j - y_k^i| < R_{max}$. En [Lowe, 1999], se propone un método de “matching” más general en el que se obtienen buenos resultados de emparejamiento para situaciones donde el análisis de vecindad no es aplicable. Dicho método considera que un emparejamiento es suficientemente fiable si la distancia entre descriptores es inferior a un límite y además dicha distancia se encuentra alejada del segundo mejor candidato de la imagen. Dicho método, aunque muy eficiente en situaciones no controladas, descarta numerosos emparejamientos válidos, debido a los posibles puntos con más de un candidato a ser emparejados. De este modo en escenas con patrones repetitivos, el método general de emparejamiento propuesto por [Lowe, 1999] descarta muchas correspondencias.

A pesar de su sencillez, el método propuesto en esta tesis, el cual elimina los descriptores pertenecientes al fondo, presenta propiedades de sumo interés:

- El método de correspondencia SIFT es invariante a la iluminación, lo que permite que el fondo utilizado como referencia pueda ser obtenido en condiciones de iluminación diferentes a la escena actual.
- Los posibles movimientos que puedan producirse en la cámara, y que cambian los parámetros extrínsecos de la misma, pueden ser detectados y corregidos mediante la correspondencia detectada entre el fondo de la escena y la nueva imagen.
- El esquema de emparejamiento propuesto permite incorporar nuevos puntos al fondo de referencia mediante un criterio de adaptabilidad. Dicho criterio no es abordado en la presente tesis, no siendo en todo caso un parámetro vital para el funcionamiento de la propuesta.

En comparación con métodos basados en la segmentación de cada píxel de la escena mediante una imagen patrón del fondo, el método propuesto presenta un alto grado de robustez ante situaciones reales, donde existen sombras, cambios de iluminación, etc...

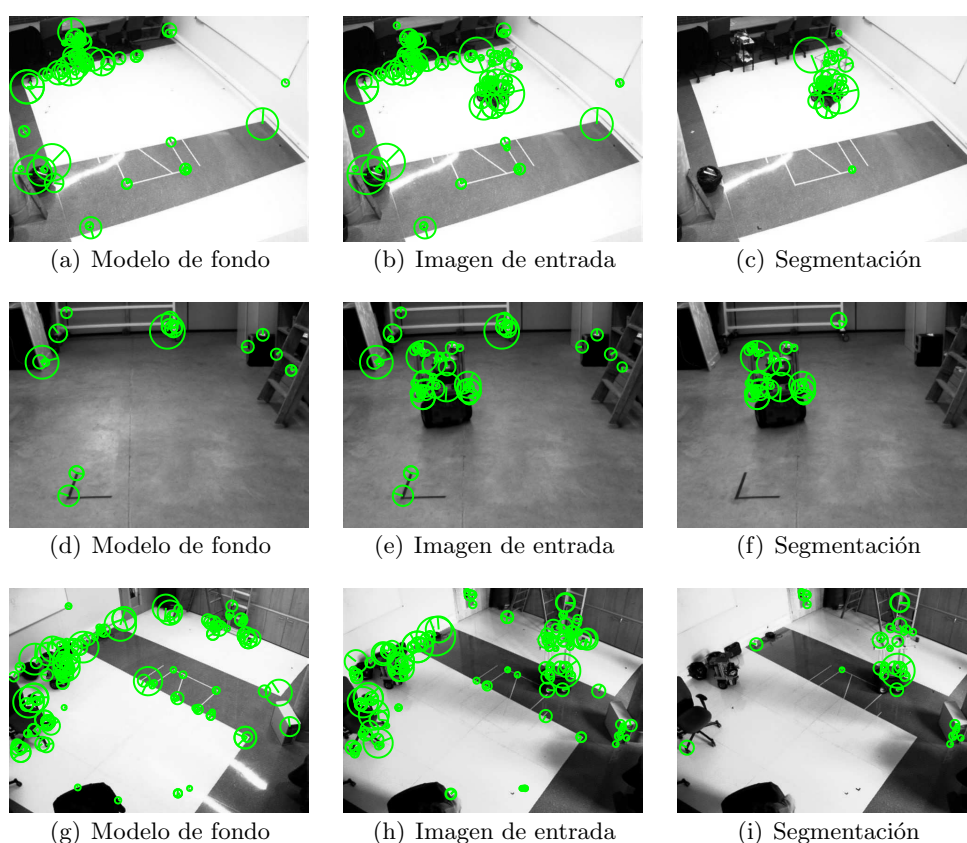


Figura 4.3: Resultado de aplicar el método de segmentación de fondo a una escena cualquiera

En la figuras 4.3.a 4.3.d y 4.3.g se muestran el conjunto de puntos de interés detectados en tres escenas diferentes utilizadas como patrón, en las figuras 4.3.b 4.3.e y 4.3.h se muestran los puntos de interés detectados en las imágenes de entrada, en la que se visualiza un

robot. En las figuras 4.3.c, 4.3.f y 4.3.i se muestran los resultados de aplicar el método de segmentación en una escena con cambios de iluminación con respecto al patrón de fondo.

Obtención de trayectorias visuales mediante un método de “tracking”

Una vez eliminados todos aquellos puntos pertenecientes al fondo, en una imagen del instante k , se obtienen conjuntos $\mathcal{C}_k = \{y_k^1, \dots, y_k^{N_k}\}$ de medidas ya filtradas. A partir de aquí, el objetivo consiste en diseñar un método eficiente para encontrar la posición de cada punto en $\mathcal{C}_k$ en los sucesivos instantes de tiempo. El método de inicialización, para el cual está diseñada la alternativa presentada en esta sección, requiere una secuencia de K instantes de tiempo en los cuales identificar diferentes puntos de la estructura del robot. De manera general se supone que la secuencia de instantes de tiempo corresponde con el intervalo temporal $k = [1, \dots, K]$. Como se hizo patente en la figura 4.2 y la tabla 4.1, los descriptores SIFT se deterioran con mucha rapidez en objetos geométricos como el propio robot que se intenta localizar. A continuación se muestra el resultado de aplicar los descriptores SIFT para obtener la posición del conjunto $\mathcal{C}_1$ en el resto de instantes de tiempo. Se muestran las trayectorias resultantes de aplicar los descriptores SIFT en dos configuraciones diferentes:

1. En la primera configuración se utilizan los descriptores $\mathcal{D}_0$ como base única para buscar las correspondencias en las sucesivas imágenes. Para mejorar la estabilidad se utiliza un “matching” guiado, del mismo modo que en el “matching” de puntos de fondo. Sea un punto en la imagen inicial y_1^i y su descriptor asociado d_1^i , entonces se propone como el punto correspondiente y_k^j en la imagen del instante k a aquel punto que cumpla las dos condiciones siguientes:

$$|y_{k-1}^l - y_k^j| < d_{max} \quad |d_1^i - d_k^j| < R_{max}, \quad (4.7)$$

siendo y_{k-1}^l el punto correspondiente a y_1^i en el instante $k - 1$. Este algoritmo pierde rápidamente la mayoría de los puntos, como se puede observar en la figura 4.4.a, si los umbrales son restrictivos y produce muchas falsas correspondencias si éstos se relajan, como se hace patente en la figura 4.4.b. Además, introduce muchas discontinuidades en las trayectorias en zonas donde existen puntos parecidos en una vecindad de los mismos. Este hecho suele ser muy común en objetos manufacturados, donde prácticamente todas las aristas y vértices presentan la misma apariencia.

2. En la segunda configuración, el procedimiento propuesto es muy similar al anterior, sin embargo, en este caso, la comparación es siempre entre conjuntos de puntos y descriptores consecutivos. Sea un punto y_k^i y su descriptor d_k^i , en la imagen actual k , entonces se considera como correspondencia al punto y_{k+1}^j en la imagen $k + 1$ si cumple las condiciones siguientes:

$$|y_k^i - y_{k+1}^j| < d_{max} \quad |d_k^i - d_{k+1}^j| < R_{max}. \quad (4.8)$$

De este modo, para averiguar la correspondencia entre el conjunto $\mathcal{C}_1$ y uno cualquiera $\mathcal{C}_k$, se debe avanzar recursivamente desde el conjunto $\mathcal{C}_2$ hasta el propio conjunto $\mathcal{C}_k$.

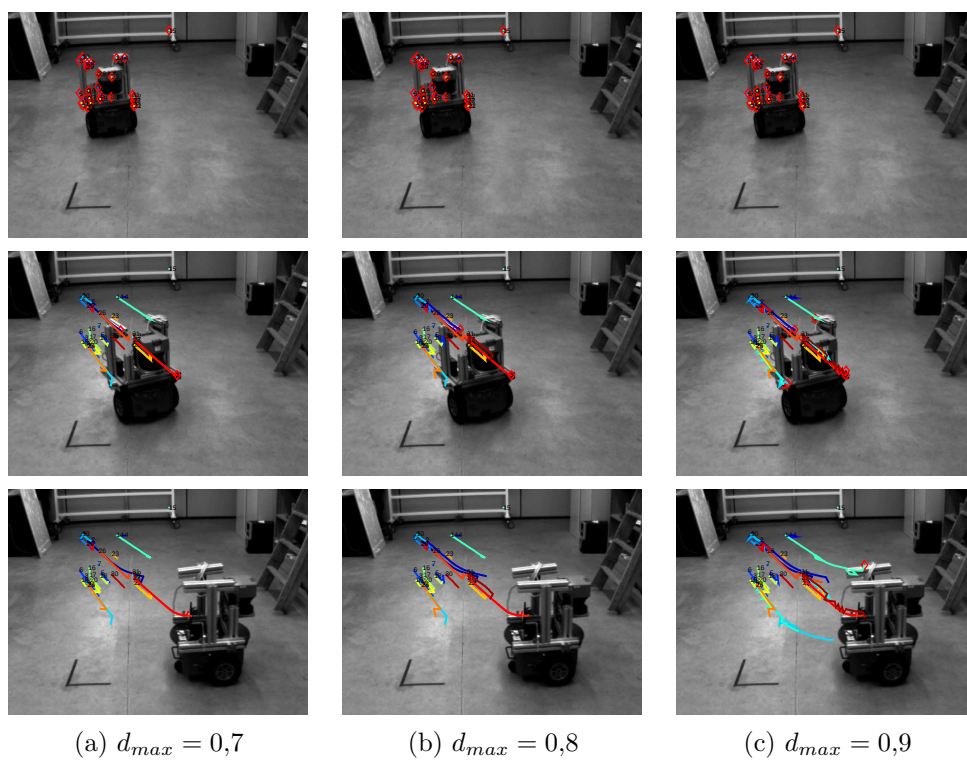


Figura 4.4: Método de “tracking” utilizando descriptores SIFT y utilizando los descriptores en $k = 1$ para obtener un “matching”.

Este desarrollo es muy similar a un proceso de “tracking”, donde se decide el emparejamiento entre dos instantes consecutivos de tiempo. Con el objetivo de robustecer el algoritmo, los descriptores SIFT utilizados pueden calcularse sin invarianza al ángulo, puesto que en imágenes consecutivas el ángulo del descriptor esencialmente no varía. Este algoritmo posee los mismos problemas de degeneración que los algoritmos de “tracking” convencionales, ya que el emparejamiento se realiza de manera incremental, teniendo en cuenta información referente a dos instantes de tiempo. Ésto produce discontinuidades en el “tracking” y muchos falsos emparejamientos. Un ejemplo de este método en la misma secuencia utilizada para evaluar el método anterior se muestra en la figura 4.5.

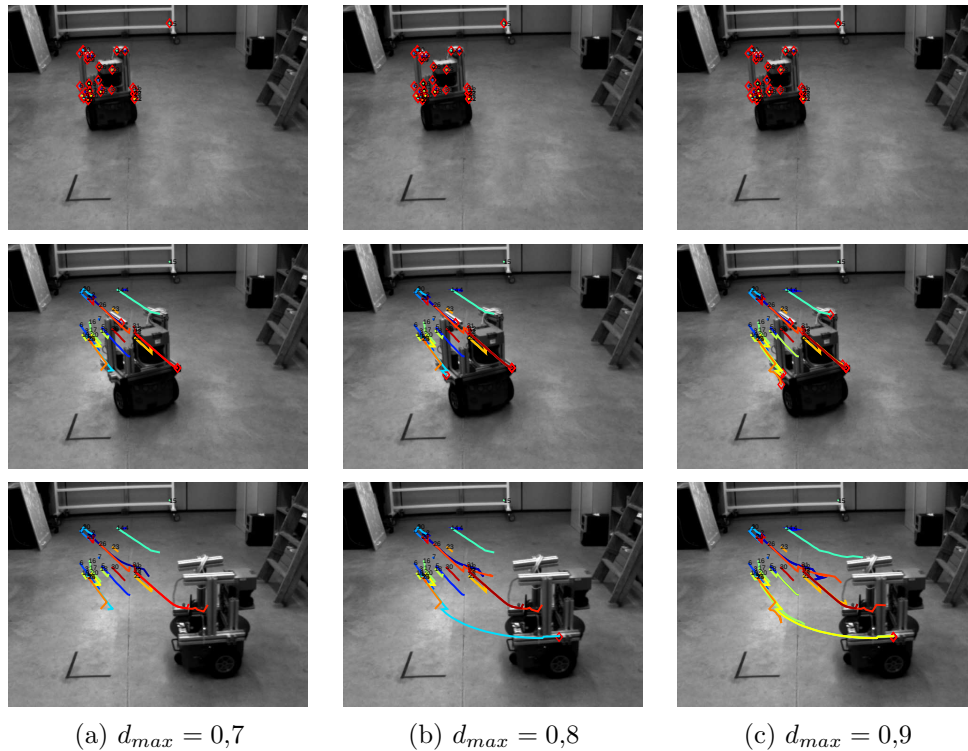


Figura 4.5: Método de “tracking” utilizando descriptores SIFT y utilizando los descriptores actualizados para obtener correspondencias.

Las alternativas anteriormente comentadas no producen secuencias suficientemente estables para ser utilizadas en el proceso de inicialización. En la mayoría de los casos las secuencias de emparejamientos son muy cortas e inestables. Para solucionar este hecho, se propone utilizar una técnica clásica de “tracking”, que en combinación con el método SIFT ofrecerá una solución al problema planteado.

Los métodos de “tracking” de puntos de interés, tales como esquinas de un objeto en movimiento, fueron desarrollados con anterioridad a la existencia de métodos “wide-baseline” de correspondencias, como lo son los descriptores SIFT. En general, si el movimiento del objeto es suficientemente pequeño, la apariencia y posición de cada punto de interés sufren una modificación limitada en imágenes consecutivas. El problema de “tracking” puede ser

abordado por esquemas muy sencillos de comparación de textura como los índices ZNCC (“Zero Normalized Cross Correlation”) o mecanismos de seguimiento como el presentado por [Shi and Tomasi, 1994] [Lucas and Kanade, 1981]. Los principales problemas de los procesos de “tracking” convencionales es la alta probabilidad de que el algoritmo entre en una situación de degeneración, por la cual un punto termina siendo identificado con otro totalmente diferente. Sin embargo, una de las ventajas de estos métodos es que son capaces de producir trayectorias visuales coherentes y estables en aquellos casos en los que no existe degeneración.

El algoritmo KLT (Kanade-Lucas-Tomasi) [Lucas and Kanade, 1981] se basa en encontrar una transformación afín (originalmente compuesta de un desplazamiento en las coordenadas (u, v) de la imagen) que mejor representa la deformación sufrida para un mismo punto de interés en dos instantes de tiempo consecutivos.

Dado un punto y_k^i se obtiene un parche $\mathcal{P}_k^i(u, v)$ que representa la textura local al punto. El tamaño de dicho parche está relacionado con dos factores: la escala a la cual el punto y_k^i es detectado y la velocidad a la que el objeto se puede desplazar entre dos imágenes consecutivas.

El algoritmo KLT encuentra la función de transformación geométrica o “warp” $(u', v') = W(u, v; q)$ que transforma, mediante los parámetros q , los parches de dos imágenes consecutivas:

$$\mathcal{P}_{k-1}^i(W(u, v; q)) = \mathcal{P}_k^i(u, v). \quad (4.9)$$

El vector q , que en general parametriza algún tipo de transformación afín, se obtiene mediante técnicas de registro de imágenes, como las comentadas en [Lucas and Kanade, 1981] o los métodos de baja complejidad presentados en [Baker and Matthews, 2004]. El punto y_{k+1}^j que corresponde a y_k^i se obtiene fácilmente una vez obtenidos los parámetros q :

$$y_{k+1}^j = W(u_{k-1}^i, v_{k-1}^i; q). \quad (4.10)$$

Mediante la repetición de dicha regla recursiva en cada instante de tiempo se obtiene una trayectoria asociada al punto y_1^i original.

Existen numerosas variantes del algoritmo KLT, y hoy en día se sigue utilizando como método para la obtención de secuencias estables de correspondencias.

En esta tesis se propone una solución al problema de la inicialización mediante un método que combina la técnica KLT con un sistema de supervisión de las situaciones degeneradas, utilizando para ello descriptores SIFT. La idea básica es aprovechar la estabilidad de los métodos KLT utilizando descriptores invariantes SIFT para supervisar el proceso de “tracking”. Mediante esta técnica se eliminarán las partes de las trayectorias que han dejado de seguir los puntos originales.

En la literatura existen propuestas que combinan, de un modo similar al propuesto en esta tesis, el tracker KLT y los descriptores SIFT propuestos por Lowe. En [Xu et al., 2007] se obtienen descriptores SIFT en ventanas de tamaño fijo para cada punto obtenido mediante el tracker KLT. En el método propuesto en esta tesis, el algoritmo KLT obtiene

no solo la posición del nuevo punto en la imagen, sino también una corrección en la escala. Dicha corrección demostrará ser de gran utilidad para proporcionar estabilidad a la propuesta de “tracking”.

A continuación se resume el método propuesto:

Se parte de un punto cualquiera, $y_k^i = (u_1^i, v_1^i)^T$, detectado en el instante k y correspondiente a la trayectoria de inicialización en la imagen $I(u, v)_k$. Dicho punto, obtenido mediante un método multiescala o como resultado de la recursión del presente método, estará consecuentemente asociado a un valor σ de escala. Mediante el método SIFT, se dispone a su vez de un descriptor d_1^i asociado a dicho punto. El valor del descriptor será, independientemente del instante de tiempo, aquel que corresponde al comienzo de la trayectoria de inicialización.

Se define un parche de textura $\mathcal{P}_k^i(u, v)$ en torno al punto y_k^i como una porción circular de textura de radio $c \cdot \sigma$:

$$\mathcal{P}_1^i(u, v) = \{I(u, v)_1 | (u - u_1^i)^2 + (v - v_1^i)^2 < c\sigma\} \quad (4.11)$$

donde $c > 3$ es un parámetro de diseño y estará relacionado con el desplazamiento máximo que puede realizar el objeto en frames consecutivos.

Se define la función $(u', v')^T = W(u, v; q)$ o “warp” como la siguiente transformación afín:

$$W(u, v; q) = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} d_u \\ d_v \end{pmatrix}, \quad (4.12)$$

siendo $q = (d_u, d_v, s)^T$, el vector de parámetros de la transformación.

Dada una imagen en el siguiente instante de tiempo $I(u, v)_{k+1}$, se plantea la obtención de los valores de q que minimizan la siguiente expresión:

$$\epsilon^2 = \sum_u \sum_v \|\mathcal{P}_{k-1}^i(W(u, v; q)) - \mathcal{P}_k^i(u, v)\|^2 \quad (4.13)$$

La expresión 4.13 se resuelve mediante un método de optimización iterativo de registro de imágenes conocido como el método de “Lucas-Kanade”. Dicho método, que requiere el cálculo de los gradientes de $\mathcal{P}_k^i$ y $\mathcal{P}_{k+1}^i$, se basa en un algoritmo de tipo Gauss-Newton para minimizar la expresión 4.13 y por tanto encontrar el valor de q .

En la figura 4.6 se muestran los resultados de las sucesivas iteraciones del método de “Lucas-Kanade”, mostrando la imagen de error obtenida como $\|\mathcal{P}_{k-1}^i(W(u, v; q)) - \mathcal{P}_k^i(u, v)\|$, para cada iteración. La suma cuadrática de los elementos de dicha imagen se corresponde con el error ϵ^2 . El valor inicial de los parámetros será $q_0 = \{0, 0, 1\}$, necesario como punto de comienzo en las iteraciones de “Lucas Kanade”.

En el algoritmo 1, se resume el esqueleto del algoritmo utilizado. En general se requieren del orden de $n < 20$ iteraciones para la obtención de los parámetros q . En [Baker and Matthews, 2004] se muestra la complejidad desglosada del algoritmo de “Lucas-Kanade”,

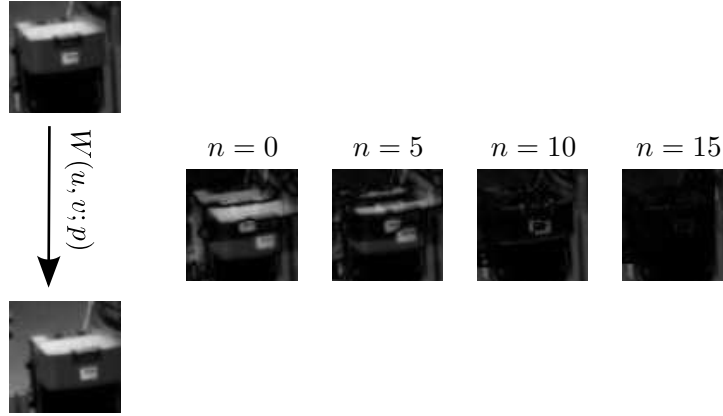


Figura 4.6: Iteraciones del algoritmo de Lucas-Kanade para un parche de textura seleccionado entre dos instantes de tiempo

siendo el paso dominante el cálculo del Hessiano de la expresión 4.19. En [Baker and Matthews, 2004] se desarrolla una versión del algoritmo que permite reducir la complejidad del problema un orden de magnitud, siendo conocido como el algoritmo “Composicional Inverso” (C.I.). En el algoritmo 2, se muestra la versión del algoritmo “Lucas-Kanade” con el método C.I. Como se muestra en la expresión 4.25 el Hessiano es siempre igual en todas las iteraciones y por lo tanto se puede precalcular con anterioridad. Las características y limitaciones del algoritmo C.I., el cual implica una aproximación, se han detallado ampliamente en [Baker and Matthews, 2004], siendo aplicado incluso a funciones de “Warp” definidas a trozos, muy utilizadas en modelos para objetos deformables. Sin embargo el algoritmo C.I. es especialmente preciso con transformaciones afines como la representada por $W(u, v; q)$ y por tanto sus resultados son en la práctica equivalentes.

Una vez finalizadas las iteraciones se obtiene el valor en el instante $k + 1$ del punto $y_{k+1}^i = W(u_k^i, v_k^i; q)$ y una redefinición de la escala $\sigma_{k+1}^i = s\sigma_k^i$, siendo s el valor del tercer elemento de los parámetros q . Mediante el nuevo valor de la posición y_{k+1}^i y la escala σ_k^i es posible obtener un valor para el descriptor SIFT en el instante $k + 1$. Los descriptores SIFT obtienen un valor de orientación característica que utilizan para corregir la textura y así obtener un descriptor invariante a la rotación. Dicho valor se obtiene automáticamente mediante la obtención de un histograma de orientaciones. En ocasiones existe la posibilidad de que el método SIFT obtenga más de una orientación dominante para un punto. En dicho caso siempre se elegirá aquel descriptor que menor distancia euclídea presenta con el obtenido para la primera iteración d_1^i .

Se propone utilizar la distancia euclídea $|d_1^i - d_k^i|$, la cual compara el descriptor SIFT en el instante inicial con el descriptor en el punto candidato, como medida de estabilidad para una correspondencia en el instante k , dada por el método propuesto. Si dicha distancia es inferior a un umbral, elegido éste poco conservativo debido a la rápida degradación de los descriptores con el movimiento del robot, se considera que el “tracker” se ha degenerado y se interrumpe el proceso para ese punto.

En el algoritmo 3 se muestra un resumen del algoritmo completo. En la práctica, como puede verse en la figura 4.7, se ha comprobado que las trayectorias obtenidas son muy

Algoritmo 1 Algoritmo Lucas-Kanade

-
- 1: **Requerimientos:**
 - 2: $I(u, v)_k$ e $I(u, v)_{k+1}$
 - 3: Punto inicial y_k^1 y su escala σ_k^i .
 - 4: Obtención de $\mathcal{P}_k^i(u, v)$ y $\mathcal{P}_{k+1}^i(u, v)$
 - 5: Inicialización del vector de parámetros q_0 y $n = 0$.
 - 6: $\epsilon^2 = \infty$, $\epsilon_{min}^2 = 0,0001$, $n_{max} = 20$
 - 8: **while** $\epsilon > \epsilon_{min}$ **and** $n < numiter_{max}$ **do**
 - 9:
 - 10: Obtención de los gradientes de $\mathcal{P}_{k+1}^i(W(u, v; q_n))$

$$\nabla \mathcal{P}^i(W(u, v; q_n))_{k+1} = \sum_l \sum_g H(u-l, v-g) \mathcal{P}^i(W(l, g; q_n))_{k+1} \quad (4.14)$$

$$H = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad (4.15)$$

- 11: Obtención de $\frac{\partial W(u, v)}{\partial q_n}$

$$\frac{\partial W(u, v)}{\partial q_n} = \begin{pmatrix} 1 & 0 & u \\ 0 & 1 & v \end{pmatrix} \quad (4.16)$$

- 12: Cálculo de $J_{\mathcal{P}} = \frac{\partial \mathcal{P}_{k+1}^i(W(u, v; q_n))}{q_n}$

$$J_{\mathcal{P}} = \nabla \mathcal{P}^i(W(u, v; q_n))_{k+1} \frac{\partial W(u, v; q_n)}{\partial q_n} \quad (4.17)$$

- 13: Jacobiano de la optimización

$$\sum_u \sum_v J_{\mathcal{P}}^T (\mathcal{P}_k^i(W(u, v; q_n)) - \mathcal{P}_{k+1}^i(u, v)) \quad (4.18)$$

- 14: Hessiano H de la optimización

$$H = \sum_u \sum_v J_{\mathcal{P}}^T J_{\mathcal{P}} \quad (4.19)$$

- 15: Actualizar el valor de $q_{n+1} = q_n - H^{-1}J$

- 16: Calcular ϵ^2

$$\epsilon^2 = \sum_u \sum_v \|\mathcal{P}_k^i(u, v) - \mathcal{P}_{k+1}^i(W(u, v; q_{n+1}))\|^2 \quad (4.20)$$

- 17: **end while**
-

Algoritmo 2 Algoritmo Lucas-Kanade mediante el método “Composicional Inverso”

- 1: $I(u, v)_k$ e $I(u, v)_{k+1}$
- 2: Punto inicial y_k^1 y su escala σ_k^i .
- 3: Obtención de $\mathcal{P}_k^i(u, v)$ y $\mathcal{P}_{k+1}^i(u, v)$
- 4: Inicialización del vector de parámetros q_0 y $n = 0$.
- 5: $\epsilon^2 = \infty$, $\epsilon_{min}^2 = 0,0001$, $n_{max} = 20$
- 6: Obtención de los gradientes de $\mathcal{P}_k^i(W(u, v; q_0))$

$$\nabla \mathcal{P}^i(W(u, v; q_0))_k = \sum_l \sum_g H(u-l, v-g) \mathcal{P}^i(W(l, g; q_0))_k \quad (4.21)$$

$$H = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad (4.22)$$

- 7: Obtención de $\frac{\partial W(u, v)}{\partial q_0}$

$$\frac{\partial W(u, v)}{\partial q_0} = \begin{pmatrix} 1 & 0 & u \\ 0 & 1 & v \end{pmatrix} \quad (4.23)$$

- 8: Cálculo de $J_{\mathcal{P}} = \frac{\partial \mathcal{P}_k^i(W(u, v; q_0))}{\partial q_0}$

$$J_{\mathcal{P}} = \nabla \mathcal{P}^i(W(u, v; q_0))_k \frac{\partial W(u, v; q_0)}{\partial q_0} \quad (4.24)$$

- 9: Hessiano H de la optimización

$$H = \sum_u \sum_v J_{\mathcal{P}}^T J_{\mathcal{P}} \quad (4.25)$$

- 10: **while** $\epsilon^2 > \epsilon_{min}^2$ **and** $n < n_{max}$ **do**

- 11:

- 12: Jacobiano de la optimización

$$\sum_u \sum_v J_{\mathcal{P}}^T (\mathcal{P}_k^i(u, v) - \mathcal{P}_{k+1}^i(W(u, v; q_n))) \quad (4.26)$$

- 13: Obtener incremento $\Delta_q = H^{-1} J$

- 14: Actualizar el valor de q mediante la regla composicional inversa

$$q_{n+1} \rightarrow W(u, v; q_{n+1}) = W^{-1}(W(u, v; q_n); q_0 + \Delta_q) \quad (4.27)$$

- 15: Calcular ϵ^2

$$\epsilon^2 = \sum_u \sum_v \|\mathcal{P}_k^i(u, v) - \mathcal{P}_{k+1}^i(W(u, v; q_{n+1}))\|^2 \quad (4.28)$$

- 16: **end while**
-

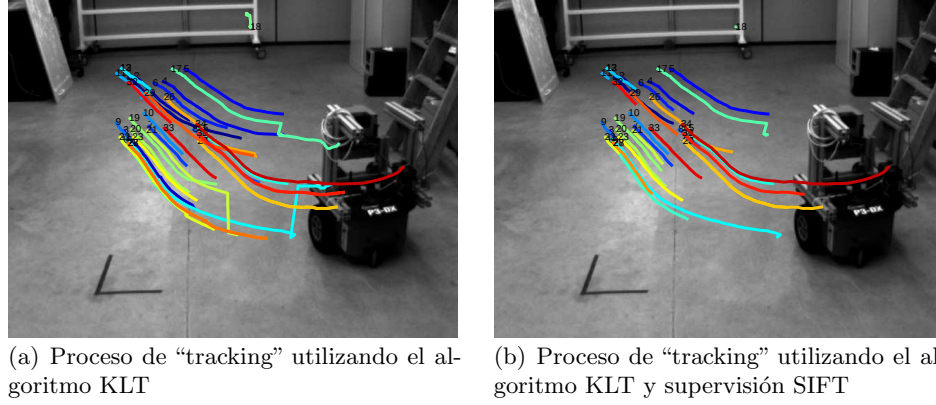


Figura 4.7: Resultado de la aplicación del algoritmo KLT con supervisión mediante SIFT a una secuencia de inicialización

estables y, si bien no están exentas de “outliers”, su número es inferior al obtenido con los otros métodos comentados. Aquellas trayectorias degeneradas que superen la criba mediante descriptores SIFT, serán identificadas y eliminadas en el capítulo 5 mediante algoritmos robustos.

Algoritmo 3 Algoritmo de “tracking” supervisado mediante descriptores SIFT

- 1: $k = 1$
 - 2: $I(u, v)_1$
 - 3: Punto inicial y_1^i y su escala σ_1^i .
 - 4: Obtención de $\mathcal{P}_1^i(u, v)$ y el descriptor asociado d_1^i
 - 5: Inicializar $dist = 0$, $dist_{max} = 0,4$
 - 6: **while** $dist < dist_{max}$ y $k < K$ **do**
 - 7: Obtención de $\mathcal{P}_{k+1}^i(u, v)$
 - 8: Aplicar algoritmo de Lucas-Kanade para encontrar y_{k+1}^i y σ_{k+1}^i
 - 9: Obtención del descriptor d_{k+1}^i asociado a y_{k+1}^i y σ_{k+1}^i .
 - 10: $dist = |d_1^i - d_{k+1}^i|$.
 - 11: **end while**
-

4.3.2. Correspondencia de marcas naturales en procesos de Estimación-Corrección

En esta sección se describe la solución al proceso de correspondencia entre la geometría del robot y las medidas en el plano imagen dentro del sistema de cálculo secuencial de la pose, el cual es detallado en el capítulo 7. El proceso secuencial de localización se basa en un enfoque de inferencia estadística, y como se verá en el capítulo 7, la solución a la localización se obtiene mediante un proceso de estimación bayesiana, resuelto por una aproximación basada en el filtro de Kalman. En esta sección se define el problema de la correspondencia dentro del propio marco probabilístico utilizado en el cálculo de la pose del robot.

A continuación se resume el proceso de correspondencia utilizado, aportando definicio-

nes previas de los procesos detallados en el capítulo 7.

Sea $\mathbf{X}_{k-1}^a$ el proceso aleatorio, con distribución de probabilidad gaussiana, que representa la pose y estructura del robot en el instante $k - 1$. Dicho proceso, cuyas componentes se definen en el capítulo 3, se puede considerar como un proceso de Markov, cuya función de densidad de probabilidad de tránsito corresponde con el modelo de movimiento descrito por la función $X_k^a = g^a(X_{k-1}^a, U_k)$. El modelo de movimiento g^a , aplicado al proceso $\mathbf{X}_{k-1}^a$, y teniendo en cuenta el ruido presente en las medidas $\mathbf{U}_k$, obtiene como resultado un nuevo proceso aleatorio $\mathbf{X}_{k|k-1}^a$, cuya varianza o incertidumbre aumenta y que corresponde con la mejor estimación disponible sin contar con las medidas actuales Y_k .

En esta tesis, el tránsito $\mathbf{X}_{k-1}^a \xrightarrow{g^a} \mathbf{X}_{k|k-1}^a$ se resuelve mediante simplificaciones de las ecuaciones integrales necesarias para su cálculo, y por tanto el carácter gaussiano de los procesos se conserva. De este modo, el proceso estimado $\mathbf{X}_{k|k-1}^a$ se representa por un vector media $\hat{X}_{k|k-1}^a$ y una matriz de covarianza $\Sigma_{k|k-1}^a$.

Del mismo modo que el tránsito del proceso $\mathbf{X}_{k-1}^a$ se realiza a través de la función g^a , dada la estimación $\mathbf{X}_{k|k-1}^a$, es posible obtener una estimación gaussiana de la proyección $\mathbf{Y}_{k|k-1}$, del conjunto de puntos que forman la estructura del robot, los cuales están contenidos en el proceso X_k^a . El proceso $\mathbf{Y}_{k|k-1}$ es consecuencia de propagar $\mathbf{X}_{k|k-1}^a$ a través de la función $Y_k = h^a(X_k^a, P)$. Al igual que en el caso anterior, se aproxima el proceso $\mathbf{Y}_{k|k-1}$ mediante una distribución gaussiana con vector media $\hat{Y}_{k|k-1}$ y una matriz de covarianza $\Sigma_{Y_{k|k-1}}$.

El proceso $\mathbf{Y}_{k|k-1}$ representa la estimación que se tiene de la posición en el plano imagen de las medidas en el instante k . El valor de $\hat{Y}_{k|k-1}$ se corresponde con las posiciones esperadas en el plano imagen de los puntos de la estructura del robot dada la pose. La matriz de covarianza $\Sigma_{Y_{k|k-1}}$ describe la incertidumbre que se tiene en la estimación de dichas medidas y la correlación de dicha incertidumbre que existe entre ellas. En general, $\Sigma_{Y_{k|k-1}}$ es una matriz completa, lo cual implica que las medidas individuales, contenidas en $\mathbf{Y}_{k|k-1}$, se encuentran correladas entre sí. Dicha correlación juega un papel importante en el proceso de estimación-corrección comentado en el capítulo 7, siendo necesaria para asegurar estabilidad en la estimación de la pose. En el proceso de “matching” que se propone a continuación tendrá también un protagonismo relevante.

Se propone un proceso de “matching” basado en la teoría de la compatibilidad estadística. Se considera al conjunto de medidas $\mathcal{C}_k$ como un conjunto en el cual se encuentra contenida una posible realización, representada como el vector Y_k , del proceso $\mathbf{Y}_{k|k-1}$. De este modo, el proceso de búsqueda de correspondencias entre $\mathcal{C}_k$ y los puntos de la estructura del robot, se soluciona desde un punto de vista de compatibilidad estadística.

Si $\mathbf{Y}_{k|k-1}$ es un proceso gaussiano de media $\hat{Y}_{k|k-1}$ y varianza $\Sigma_{Y_{k|k-1}}$, entonces:

$$\mathbf{d}_{\text{mah}} = (\hat{Y}_{k|k-1} - \mathbf{Y}_{k|k-1})^T \Sigma_{Y_{k|k-1}}^{-1} (\hat{Y}_{k|k-1} - \mathbf{Y}_{k|k-1}), \quad (4.29)$$

es un proceso aleatorio $\chi^2(N_Y)$, con N_Y grados de libertad. Siendo N_Y la longitud de Y_k . La expresión (4.29) se conoce como distancia de Mahalanobis, si es aplicada a una realización concreta del proceso $\mathbf{Y}_{k|k-1}$.

De este modo si Y_k es una posible realización de dicho proceso, la compatibilidad estadística entre ambos, o de otro modo, la confianza con la que Y_k representa una realización de $\mathbf{Y}_{k|k-1}$ será:

$$d_{mah} = (\hat{Y}_{k|k-1} - Y_k)^T \Sigma_{Y_{k|k-1}}^{-1} (\hat{Y}_{k|k-1} - Y_k), \quad (4.30)$$

donde el valor de d_{mah} puede ser contrastado con la distribución acumulativa χ^2 para obtener un umbral que permita asegurar hasta un nivel de confianza prefijado, el 90 % usualmente, que la realización Y_k es compatible con el proceso $\mathbf{Y}_{k|k-1}$.

El algoritmo planteado en esta tesis, que da solución al problema de “matching”, se basa en encontrar el vector de correspondencias Y_k entre los elementos del conjunto $\mathcal{C}_k$ que minimiza la distancia de Mahalanobis. Cada posible combinación del vector Y_k da lugar a un valor correspondiente de d_{mah} , relacionado directamente con su índice de confianza de pertenecer al proceso $\mathbf{Y}_{k|k-1}$. Debido a que la matriz $\Sigma_{Y_{k|k-1}}$ es completa, la combinación de elementos del conjunto $\mathcal{C}_k$ que forman el vector de medidas Y_k óptimo, requiere evaluar un número elevado de combinaciones. Esto se debe al hecho de que no es posible saber si una componente del vector Y_k es óptima, sin conocer la influencia que tiene el resto a través de los términos de correlación mutua de la matriz de covarianza. El uso de la matriz de covarianza completa en la expresión (4.30) se denomina test de compatibilidad conjunta, y sus propiedades son estudiadas en [Neira and Tardos, 2001].

Debido a la complejidad que requiere el uso de una matriz de covarianza completa, la matriz $\Sigma_{Y_{k|k-1}}$ se suele simplificar por una diagonal por bloques, representando cada bloque la covarianza asociada a las dos coordenadas de cada punto independiente $\Sigma_{Y_{k|k-1}}^i$ y contenido en $\mathbf{Y}_{k|k-1}$:

$$\Sigma_{Y_{k|k-1}} \approx \begin{pmatrix} \Sigma_{Y_{k|k-1}}^1 & \cdots & \bar{0} \\ \vdots & \ddots & \vdots \\ \bar{0} & \cdots & \Sigma_{Y_{k|k-1}}^N \end{pmatrix}. \quad (4.31)$$

El hecho de suponer una matriz diagonal, permite encontrar el mínimo de (4.30) mediante un test de compatibilidad individual por cada elemento del proceso $\mathbf{Y}_{k|k-1}$ empleando su correspondiente bloque individual de covarianza. Ésta técnica se denomina test de compatibilidad individual y es comúnmente utilizada por su simplicidad en procesos de estimación-corrección.

La simplificación realizada por el test de compatibilidad individual, como se mostrará con un ejemplo sencillo, tiene consecuencias directas en la robustez con la que es capaz de detectar la combinación de Y_k que minimiza la expresión (4.30).

En la figura 4.8 se muestra un ejemplo simulado del proceso de “matching” basado en ambos test de compatibilidad comentados. En la figura 4.8.a se muestra un conjunto de $N = 4$ puntos, dispuestos según la pose X_k y vistos por una cámara. En la figura 4.8.b se muestra un conjunto de medidas correspondientes a los vértices de dicho polígono $y_k^1, \dots, y_k^4$ y una medida adicional y_k^5 debida a un posible fallo en el proceso de detección. El conjunto de medidas $\mathcal{C}_k$ contiene por tanto cinco valores de medida. El proceso $\mathbf{Y}_{k|k-1}$,

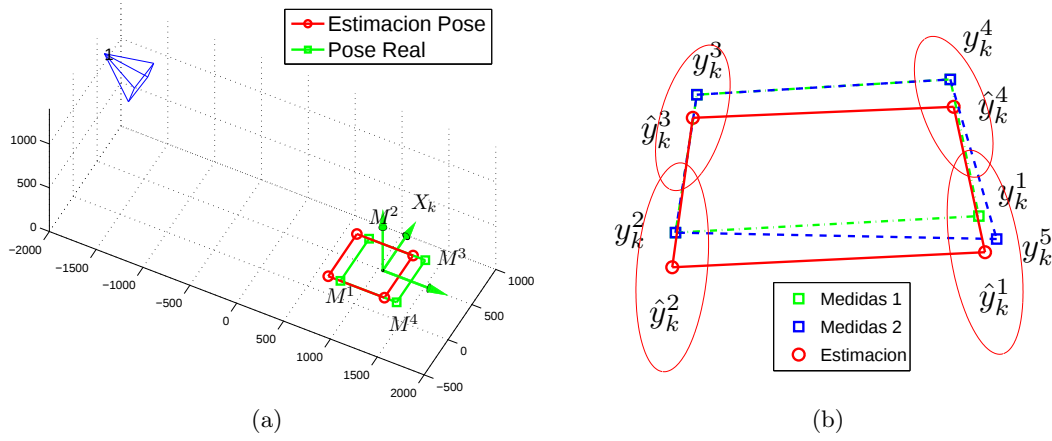


Figura 4.8: Ejemplo correspondiente a un polígono de cuatro puntos proyectado en una cámara.

presenta un vector media $\hat{Y}_{k|k-1}$ que es representado en la figura 4.8 mediante los valores $\hat{y}_k^1, \dots, \hat{y}_k^4$. Cada bloque de covarianza individual, que componen la versión simplificada de la varianza del proceso $\Sigma_{Y_{k|k-1}}$, se muestra en la figura 4.8.b como una elipse que encierra el 90 % de la distribución gaussiana equivalente. De este modo, una medida es compatible individualmente con su correspondiente estimación si se encuentra contenida en la elipse. En la tabla 4.2 se muestra el resultado de aplicar ambos test de compatibilidad a dos posibles combinaciones para el vector Y_k .

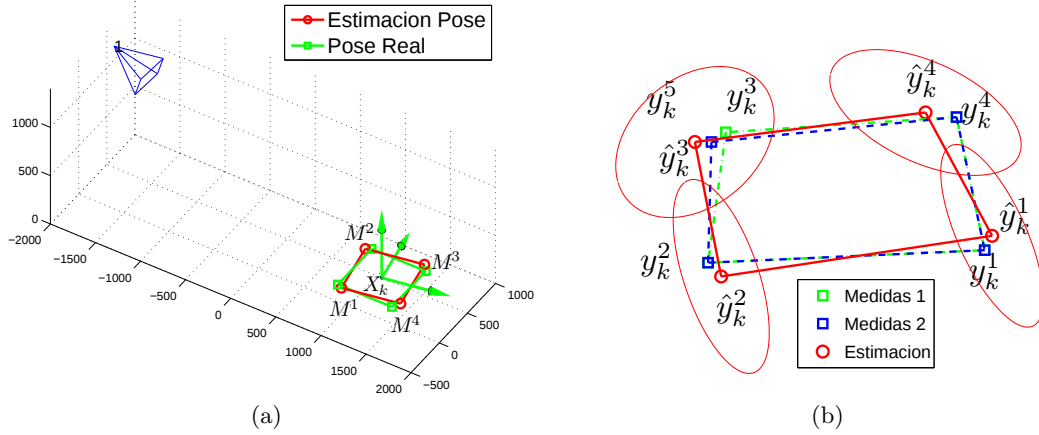
Como se observa en la citada tabla, la compatibilidad individual acepta como medida más cercana a y_k^5 , que proviene de un error de detección, pero es más cercana “individualmente” a uno de los elementos del proceso $Y_{k|k-1}$. Sin embargo, utilizando un test de compatibilidad conjunta, la combinación que contiene la asociación errónea está más alejada “conjuntamente” que la combinación correcta. De hecho, el valor de compatibilidad para la combinación que contiene y_k^5 como medida no es compatible al ser su valor de distancia superior al indicado para una probabilidad del 99 %.

Combinación	Compt. Indv	Compt. Conjunta	Compatibilidad 99 %
$y_k^1, y_k^2, y_k^3, y_k^4$	11.3427	9.4662	20.1
$y_k^5, y_k^2, y_k^3, y_k^4$	9.7417	81.6748	20.1

Tabla 4.2: Tabla que muestra la compatibilidad individual y conjunta para el ejemplo propuesto.

En la figura 4.9 se muestra la misma situación cuando se ha producido una rotación en el grupo de cuatro puntos y existe una medida y_k^5 más cercana a la estimación $\hat{y}_k^3$. El test individual elegiría la medida errónea como la correspondencia más probable, mientras que el test de compatibilidad conjunta es capaz de discernir que la medida y_k^3 está correlada con el resto.

El test de compatibilidad utilizando la matriz completa requiere, como se ha comentado con anterioridad, evaluar múltiples combinaciones de las medidas para averiguar cual de ellas equivale a un mínimo en el test de Mahalanobis.



Combinación	Compt. Indv	Compt. Conjunta	Compatibilidad 99 %
$y_k^1, y_k^2, y_k^3, y_k^4$	9.0774	6.5047	20.1
$y_k^1, y_k^2, y_k^5, y_k^4$	7.1577	53.4079	20.1

Figura 4.9: Ejemplo correspondiente a un polígono de cuatro puntos proyectado en una cámara.

Este hecho ha motivado la búsqueda en la literatura de algoritmos que sean capaces de implementar el test de compatibilidad conjunta en un tiempo razonable. Existen derivaciones del mismo que se basan en búsqueda eficiente de grafos [Neira and Tardos, 2001] propuestos como solución al problema del “matching” en procesos de estimación-corrección.

En esta tesis se plantea la utilización de un test de compatibilidad para obtener en cada instante de tiempo un vector de medida Y_k correspondiente a la estructura del robot. Sin embargo, a pesar de las bondades del test de compatibilidad conjunta, y del hecho de que existan algoritmos relativamente eficientes para resolverlo, en esta tesis se plantea utilizar el test de compatibilidad individual como base para encontrar correspondencias en procesos de estimación corrección.

Como se detalla en el capítulo 7, se implementa un algoritmo robusto, basado en el método RANSAC, que es muy eficiente en terminos de tiempo y exacto eliminando medidas erróneas que pudieran ser incluidas en el vector Y_k debidas a la simplificación. De este modo, cada medida individual, asociada a su correspondiente matriz de covarianza individual, no será admitida si no supera un umbral de confianza, determinado mediante su medida individual de d_{mah} .

En el capítulo 7 se mostrarán resultados del método propuesto, ya integrado dentro del proceso de estimación-corrección.

4.3.3. Correspondencias “Wide-Baseline”

El problema que se aborda a continuación consiste en la obtención de correspondencias entre puntos del modelo tridimensional del robot y una imagen del mismo en la que la pose es desconocida. Asociado a cada punto del modelo tridimensional, se dispone de un modelo de apariencia, basado en este caso en descriptores SIFT, el cual debe permitir obtener

correspondencias en imágenes de las que se desconoce la pose del robot. Este esquema es conocido como “Matching by Detection” y ha sido comentado en el capítulo 2. En general, este tipo de técnicas buscan la pose del robot a partir de correspondencias y por tanto el método de “matching” se combina con un algoritmo robusto que permite eliminar medidas erróneas o “outliers”.

Los métodos de medida que se han comentado con anterioridad, correspondientes al proceso de inicialización y al algoritmo secuencial, aunque centrados en solucionar igualmente el “matching”, se diferencian del problema que se trata a continuación en la posibilidad de contar con una estimación de movimiento. Dicha estimación consiste en la suposición de movimiento suave del proceso de inicialización o la estimación de movimiento mediante odometría en el proceso de estimación secuencial de la pose.

El enfoque propuesto en este punto no posee estimación de pose y por tanto el método de “matching” se enfrenta al problema completo de correspondencias “wide-baseline”, que consiste en encontrar correspondencias ante cualquier cambio de perspectiva dado por diferentes poses del robot. Este problema, dentro de una propuesta de localización de robots, se denomina secuestro o reinicialización del sistema, y es causado por una pérdida en la estimación secuencial de la pose. Gracias al método de emparejamiento que se comenta en esta sección, el problema del secuestro admite una solución sencilla, la cual se describe en el capítulo 7.

Los descriptores SIFT presentan características ideales que permiten solucionar el problema planteado, ya que mediante un esquema de emparejamiento, como el propuesto por [Lowe, 1999], es posible encontrar correspondencias de un conjunto de descriptores en cualquier imagen. Sin embargo, como se ha detallado al comienzo del capítulo, existe una rápida degradación de los descriptores SIFT ante movimientos del robot, puesto que los puntos de su estructura no cumplen en general la suposición de que un entorno de los mismos está contenido en un plano. Utilizando un solo descriptor SIFT como modelo de apariencia de cada punto M^i , en casos reales no es estable. Las correspondencias mediante el uso de un descriptor se encuentran limitadas a movimientos en torno a la pose en la que se obtuvo el conjunto de descriptores de referencia, lo cual limita enormemente la posibilidad de su utilización en el problema del secuestro.

Como solución a la rápida degeneración de descriptores SIFT en objetos no planos, se plantea la inclusión de un modelo de apariencia asociado a cada punto M^i , que permita identificar adecuadamente la variabilidad de los descriptores resultante de los cambios de pose en el robot. Dicho modelo de apariencia, se construye a partir de un conjunto representativo de descriptores SIFT por cada punto del robot. La obtención de dicho conjunto, a modo de fase de entrenamiento, se realiza en cualquiera de las fases anteriormente comentadas, ya sea usando para ello las medidas filtradas por el algoritmo de Lucas-Kanade o medidas libres de “outliers” en la fase secuencial del algoritmo.

De este modo, se supone que existe un conjunto de N_d^i descriptores $\mathcal{D}^i = \{d_1^i, \dots, d_{N_d^i}^i\}$ asociados a un punto M^i del modelo tridimensional del robot y resultado del entrenamiento. Cada conjunto $\mathcal{D}^i$ poseerá una cantidad de información que difiere en función de la estructura geométrica a la que pertenece el punto M^i . Si dicho punto forma parte de un plano, la precisión con la que una transformación afín modela las deformaciones locales, debidas a cambios de perspectiva, será suficientemente precisa, siendo por tanto esperable

que el conjunto de descriptores asociados sean muy cercanos entre sí. Por el contrario, si el punto M^i pertenece a una esquina del objeto, los cambios en el fondo y el hecho de que una esquina no es localmente plana, producirán un conjunto de descriptores SIFT muy diferentes.

En esta tesis se plantea utilizar un análisis de las componentes principales (PCA) aplicado a cada conjunto $\mathcal{D}^i$. Mediante esta técnica es posible no solo reducir la dimensión de aquellos conjuntos con elementos similares, sino también eliminar aquellas partes de los descriptores que son debidas a distorsiones con una variabilidad muy alta, como ocurre cuando representan una esquina o puntos con fondo cambiante.

El algoritmo de reducción de la dimensionalidad mediante PCA, así como la definición de un proceso de “matching” se comentan a continuación:

Sea el conjunto de descriptores $\mathcal{D}^i$, asociados al punto M^i , donde cada descriptor $d_j^i \in \mathbb{R}^{128}$ es un vector columna de 128 elementos. Se compone la matriz L como la concatenación del conjunto de descriptores:

$$L = \begin{pmatrix} d_1^i - \hat{d}^i & \cdots & d_{N_d^i}^i - \hat{d}^i \end{pmatrix} \quad \hat{d}^i = \frac{1}{N_d^i} \sum_{j=0}^{N_d^i} d_j^i \quad (4.32)$$

A partir de la matriz L , donde cada columna representa a cada descriptor menos la media del conjunto $\hat{d}^i$, se obtiene la siguiente descomposición en valores singulares:

$$LL^T = WDW^T, \quad (4.33)$$

donde W es una matriz ortogonal y D es una matriz diagonal, que contiene autovalores siempre positivos o nulos. Se denominan W_t y D_t a las matrices W y D truncadas hasta un número de autovalores concreto, estableciendo para ello un umbral mínimo d_{min} que permite así filtrar aquellas “dimensiones” o direcciones de los descriptores debidos a elementos con alta variabilidad. La determinación del umbral d_{min} se realiza en función del conjunto completo de autovalores de la matriz D . Si se denomina d_s a la suma de todos los autovalores de LL^T , se establece el umbral d_{min} como un porcentaje dado de d_s . En la práctica, se ha observado que un umbral de $d_{min} = 0,7d_s$ ofrece los mejores resultados con el tipo de imágenes que resultan del robot en movimiento.

La matriz W_t define un cambio de base entre el espacio de 128 elementos de los descriptores a un espacio de dimensión reducida, dada por el número de autovalores que permanecen en la matriz D_t . De este modo, dado un descriptor cualquiera d , su transformación en la base será:

$$Y_D^i = W_t(d - \hat{d}^i) \quad (4.34)$$

Mediante la matriz W_t^T y la media $\hat{d}^i$ se puede reconstruir un vector $\tilde{d}$ de 128 componentes:

$$\tilde{d} = W_t^T Y_D^i + \hat{d}^i = W_t^T W_t(d - \hat{d}^i) + \hat{d}^i \quad (4.35)$$

Dicho vector $\tilde{d}^i$ representa la reconstrucción en el espacio de 128 coordenadas del vector Y_D^i y por tanto equivale al descriptor más cercano a d^i , generado por la base W_t obtenida mediante PCA.

Se plantea utilizar el descriptor $\tilde{d}^i$ y su distancia al descriptor d como medida de “matching” entre una base generada para un punto M^i del modelo tridimensional del robot y el descriptor d . Como medida de distancia se utiliza la distancia euclídea entre $\tilde{d}^i$ y el vector d :

$$w = |d^T - \tilde{d}^i| \quad (4.36)$$

De este modo, dadas las matrices $W_t^1, \dots, W_t^N$ y $\hat{d}^1, \dots, \hat{d}^N$, correspondientes a los puntos del robot $M^1, \dots, M^N$, el proceso de emparejamiento con un conjunto de descriptores $d_k^1, \dots, d_k^{N_c}$, obtenidos en la imagen I_k se muestra en el algoritmo 4. El criterio utilizado se basa en encontrar el descriptor de entrada d que tiene la menor distancia para cada base correspondiente a los puntos del robot, definida en (4.36).

Algoritmo 4 Proceso de emparejamiento utilizando SIFT + PCA

```

1: Requerimientos
2: Bases PCA de los N puntos  $W_t^1, \dots, W_t^N$  y  $\hat{d}^1, \dots, \hat{d}^N$ 
3: Descriptores de la imagen de entrada  $d_k^1, \dots, d_k^{N_c}$ .
4: Puntos en el plano imagen  $y_k^1, \dots, y_k^{N_c}$ .
5: while  $j < N_c$  do
6:   while  $i < N$  do
7:     Vector reconstruido  $\tilde{d}^i = W_t^T Y_D^i + \hat{d}^i = W_t^T W_t (d_k^j - \hat{d}^i) + \hat{d}^i$ 
8:      $w(i) = |(d_k^j) - \tilde{d}^i|$ 
9:   end while
10:  Ordenación descendente  $w_s = w(ind)$  del vector  $w$ .
11:  1º y 2º mejores emparejamientos:  $w_s(1), i_1 = ind(1)$  y  $w_s(2), i_2 = ind(2)$ 
12:  if  $w_2 > 1,5w_1$  then
13:     $M^{i_1}$  está emparejado con el punto  $y_k^{j_1}$  mediante el correspondiente  $d_k^{j_1}$ 
14:  end if
15:   $i = i + 1$ 
16: end while

```

Al igual que en la propuesta de Lowe, se descartan aquellos emparejamientos en los cuales el descriptor de la imagen de entrada es también muy cercano a otro punto del modelo tridimensional del robot. En este caso se impone un criterio sencillo, en el que se busca que el mejor emparejamiento tenga una distancia 1,5 veces menor que el segundo mejor como es descrito en [Lowe, 1999].

Con el objetivo de validar el método propuesto, se propone un experimento realizado en condiciones reales, en el cual se observa un robot en una escena con una cámara. Se utilizará una trayectoria de inicialización, mediante la cual se obtendrá el modelo tridimensional de $N = 10$ puntos del robot, el cual se representa en la figura 4.10.b y proyectados en la figura 4.12. En la figura 4.10.a se muestra la disposición geométrica de la cámara con respecto a la trayectoria utilizada para obtener el modelo tridimensional del robot, así como los descriptores asociados a cada uno. La secuencia de entrenamiento o inicialización, consistente en 100 instantes de tiempo, se submuestra para obtener 50 instantes de tiempo,

a partir de los cuales se obtienen como máximo 50 descriptores asociados a cada punto.

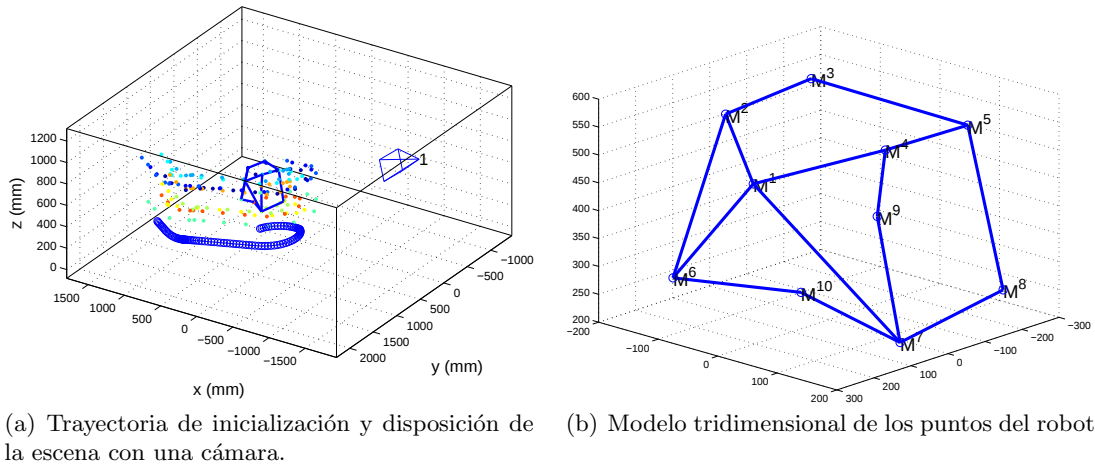


Figura 4.10: Representación de la trayectoria de inicialización utilizada para realizar el entrenamiento de los descriptores y el modelo tridimensional del robot

En la figuras 4.13.a y 4.13.b, se muestran respectivamente los resultados de aplicar el método utilizando un solo descriptor como modelo de apariencia, en comparación con el método propuesto a una trayectoria extensa que se muestra en la figura 4.11. En las figuras 4.13.a y 4.13.b se muestran los resultados del “matching”, diferenciando correspondencias correctas o “inliers” del total de correspondencias. Para realizar dicha diferenciación, se ha utilizado un método robusto, basado en RANSAC, que busca la pose consensuada del robot a partir de las correspondencias totales encontradas. El método utilizado para calcular la pose será comentado en el capítulo 6 y su aplicación con RANSAC para resolver el problema del secuestro en el capítulo 7.

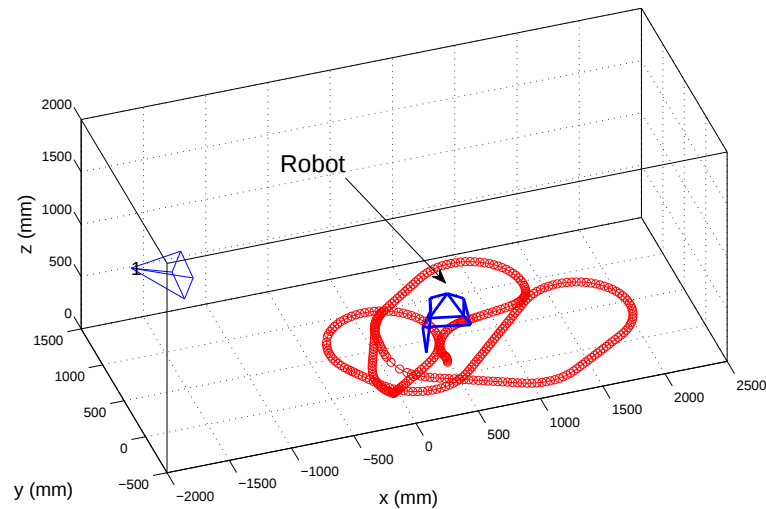


Figura 4.11: Trayectoria de prueba del método de matching

Como se muestra en la figura 4.13.a, el método SIFT con un solo descriptor pierde

los emparejamientos rápidamente, mientras que el método basado en PCA, en la figura 4.13.b, es capaz de mantener un conjunto de correspondencias estables. En la figura 4.12 se muestran las imágenes utilizadas para realizar el entrenamiento de los descriptores SIFT y en la figura 4.14 se muestra el “matching” resultante en ciertos instantes de tiempo seleccionados y mostrados en el pie de la figura.

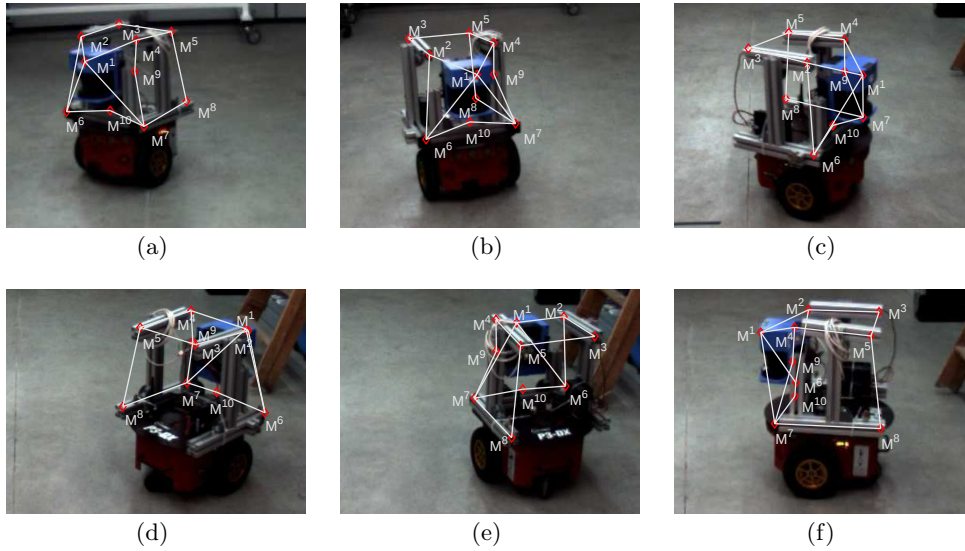


Figura 4.12: Entrenamiento

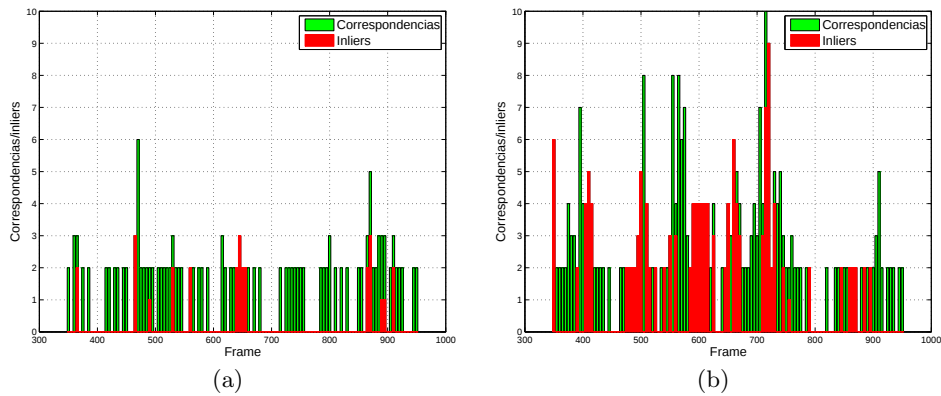


Figura 4.13: Numero de correspondencias totales (verde) e “inliers” (rojo) para cada instante de tiempo de la trayectoria de prueba

4.4. Conclusiones

En este capítulo se ha presentado el proceso de medida utilizado, desglosado según las diferentes situaciones y requerimientos que resultan del sistema completo de localización propuesto en la tesis. La aportación más notable que se realiza en este capítulo son las

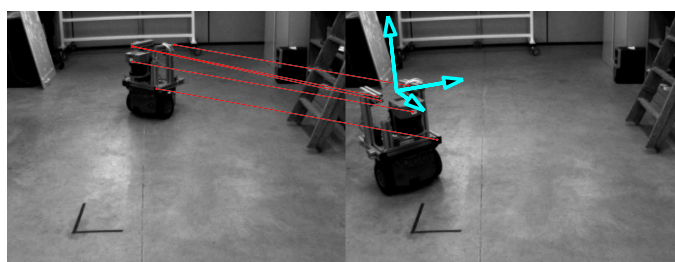
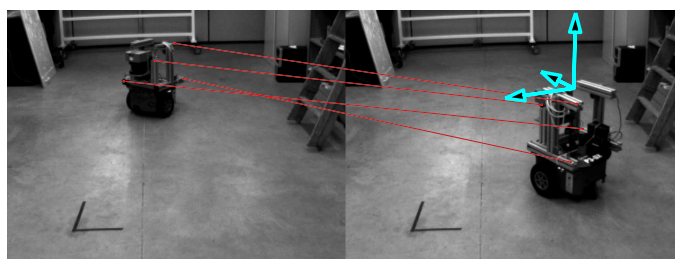
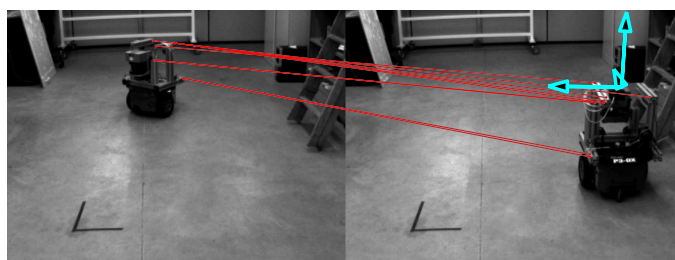
(a) $k = 500$ (b) $k = 605$ (c) $k = 720$

Figura 4.14: Resultado del “matching” para imágenes seleccionadas de la trayectoria de prueba

propuestas realizadas para cada problema planteado, según la información disponible y los requerimientos. Se ha demostrado que es necesario recurrir a una sinergia entre diferentes metodologías, que si bien son ampliamente conocidas en la literatura, su aplicación en la problemática presentada es novedosa.

Las aportaciones principales de este capítulo se resumen a continuación:

- Se ha demostrado la debilidad de los descriptores SIFT en procesos de seguimiento de objetos tridimensionales, en los que el modelo de deformación afín local a cada punto no es válido. Se han propuesto alternativas de “matching” guiado que dada la rápida degradación de los descriptores producen una serie de trayectorias visuales insuficientes para el proceso de inicialización.
- Se ha propuesto un sistema de “tracking” que combina el algoritmo KLT, el cual demuestra ser estable en objetos tridimensionales, con descriptores SIFT para evitar la degeneración propia del algoritmo KLT.
- Se ha propuesto un sistema de emparejamiento “Wide-Baseline” para objetos tridimensionales utilizando para ello una base de entrenamiento obtenida durante las fases de inicialización y el proceso secuencial de cálculo de la pose. Mediante una reducción de dimensionalidad utilizando PCA se ha obtenido un sistema que incrementa notablemente el número de emparejamientos correctos con respecto al uso de descriptores SIFT convencionales.
- Todas las propuestas realizadas han sido validadas con experimentos prácticos y se encuentran integradas en el proceso de localización que se desarrolla en esta tesis.

Capítulo 5

Inicialización de la pose y geometría

5.1. Introducción

En el presente capítulo se aborda el problema de la inicialización del vector de estados, es decir, la obtención del valor de X_0^a , compuesto por pose y estructura, en el primer instante en el que se observa un nuevo robot. Por motivos de claridad en la notación, el instante de tiempo correspondiente a la inicialización se identifica con $k = 0$, y así será expuesto a lo largo del capítulo. En el caso de existir varios robots, cuya aparición puede ocurrir en momentos diferentes de tiempo, el instante de tiempo inicial no corresponderá con $k = 0$, sin embargo, las conclusiones y el proceso serán equivalentes a las expuestas en este capítulo.

El problema propuesto de inicialización presenta una dificultad añadida dada la carencia de información que permita una estimación directa de X_0^a . En $k = 0$ se tiene como fuentes de información acerca de un robot nuevo los siguientes elementos:

1. Capacidad para ejercer control sobre el robot a través de la red de comunicaciones.
2. Lecturas de odometría obtenidas de los sensores locales del robot.
3. Medidas en el plano imagen, a modo de marcas naturales, incluyendo información de las zonas de la imagen correspondiente a objetos en movimiento.

En esta tesis se propone que cada robot de reciente aparición en el entorno realice un movimiento de inicialización o “calibración” que permita que el “Espacio Inteligente” obtenga información del mismo.

Las medidas de odometría, aún tratándose de información precisa sobre el movimiento del robot en trayectorias cortas, ofrecen una estimación siempre relativa a una posición inicial X_0 y por tanto, desconociendo dicha información, no son suficientes para obtener la pose del robot. Las medidas en el plano imagen, dadas las características de la función de proyección, no pueden ser utilizadas para calcular la pose del robot, en ausencia de otras fuentes de información adicionales.

Bajo estas condiciones se propone un algoritmo de estimación de la pose inicial X_0^a . En este capítulo se aborda dicho problema fundamental ante dos circunstancias que exigen diferentes planteamientos:

- En primer lugar se trata el problema de la obtención de la pose cuando el robot es visible por una sola de las cámaras. En este primer caso, la pose y la estructura inicial del robot se obtienen mediante un proceso de reconstrucción a partir de medidas tomadas en el plano imagen usando la única cámara disponible. Dichas medidas serán correspondientes a una trayectoria realizada por el robot y que permite generar suficiente información para obtener un conjunto de puntos tridimensionales de su estructura, así como el valor de su pose. La ambigüedad natural, debida al uso de una cámara, permite obtener una reconstrucción del robot definida en todos sus parámetros salvo un factor de escala. Dicho factor de escala necesita de una referencia métrica, independiente de las medidas en la cámara, que traslada la solución a su escala correcta. En esta tesis se propone el uso de los sensores de odometría como estimación métrica del movimiento del robot.

Sin embargo, dadas las propiedades de error acumulativo presentes en el posicionamiento por odometría, su uso puede dar lugar a importantes errores en la solución buscada. Se propone, por tanto, un sistema de reconstrucción que incluya entre sus propiedades la capacidad de ponderar adecuadamente la influencia de dicho error en la pose y la geometría obtenida.

- En segundo lugar, se tratará el caso del uso de varias cámaras para inicializar la pose del robot. En este caso, la referencia métrica provendrá del conocimiento de la configuración geométrica entre las distintas cámaras. El algoritmo, al igual que en el caso de una sola cámara, tomará medidas de puntos pertenecientes a la geometría del robot en las respectivas imágenes. Dada la configuración de cámaras separadas o “wide-baseline”, la información en cada uno de los planos imagen es difícil de correlar directamente. Por ello se propone evitar correspondencias entre cámaras, y emplear la fusión de información directamente en términos de la pose del robot. Este enfoque permite que cada cámara, en principio, obtenga su propia estimación independiente de la geometría del robot. Un paso posterior, dedicado a refinar la solución obtenida, fundirá aquellos puntos reconstruidos desde distintas cámaras, que pertenecen al mismo punto geométrico, y formará con el conjunto el vector de estados inicial X_0^a buscado.

5.2. Inicialización de la pose y geometría del robot utilizando una única cámara

La motivación de esta sección es proponer una solución, mediante una sola cámara, al problema de la inicialización de la pose y la geometría que componen el vector aumentado X_0^a del robot. Dada la imposibilidad de obtener información tridimensional mediante una sola imagen, se plantea una propuesta que implica la realización de una trayectoria de inicialización por parte del robot, la cual es captada por la única cámara disponible. Mediante un conjunto de imágenes del robot en movimiento y la información proveniente de la odometría, integrada en el algoritmo como referencia métrica, se obtiene la “reconstrucción” del vector X_0^a . El término “reconstrucción” será utilizado a este capítulo puesto que el vector X_0^a requiere la obtención no sólo de la pose del robot sino también de un modelo tridimensional del mismo.

La reconstrucción de una escena, en este caso la pose del robot y su estructura, a partir de medidas realizadas por una cámara es un problema clásico en visión por computador. Entre el amplio conjunto de propuestas, destacan aquellas técnicas agrupadas bajo el nombre de “Bundle Adjustment” [Triggs et al., 2000] [Hartley and Zisserman, 2003] y cuyo cometido esencial es estimar, de manera conjunta y óptima, la estructura 3D de una escena y los parámetros de cámara de un conjunto de imágenes tomadas de la misma. El perfeccionamiento de dichas técnicas hace posible la reconstrucción de escenas consistentes en miles de puntos y vistas por cientos de cámaras en un tiempo del orden de segundos con el “hardware” actual.

En términos generales, la reconstrucción a partir de información visual se resuelve mediante un proceso iterativo que busca optimizar una medida de discrepancia entre las

magnitudes medidas en la imagen, con respecto a lo que se espera medir, dada una estimación de X_0^a . La medida de discrepancia permite, tanto asegurar que todas las medidas contribuyen a la solución final de una manera justa, atendiendo a su incertidumbre, como evitar la inclusión de “outliers” o espurios en el problema.

En este punto se propone reconstruir el vector X_0^a a partir de medidas en el plano imagen y la información de los sensores de odometría utilizando para ello una técnica de “Bundle Adjustment”.

El conocimiento estadístico y detallado del problema visto en el capítulo 3 permite generar una medida de discrepancia que atenúa la influencia negativa que tiene el error acumulativo de la odometría en la solución final de X_0^a . Para ello se plantea utilizar un criterio de Máxima Verosimilitud que relaciona medidas y el valor de X_0^a , utilizando una función de densidad de probabilidad (f.d.p.) como medida de discrepancia.

La contribución que realiza la tesis en este respecto es la inclusión de todo el modelado estadístico del problema para generar una versión realista de la f.d.p. utilizada para medir la verosimilitud. Si se identifica por $\mathbf{Y}_{\mathcal{T}}$ al proceso que representa a todas las medidas obtenidas en la secuencia de inicialización, se buscará el máximo de distribución $p(\mathbf{Y}_{\mathcal{T}}|X_0^a)$ en función de X_0^a . Dicho enfoque garantiza que las contribuciones a la solución final por parte de la información, tengan un peso final en X_0^a que depende de su verosimilitud o en otras palabras de la varianza calculada para ellas mediante la distribución $p(\mathbf{Y}_{\mathcal{T}}|X_0^a)$, la cual será determinada en gran medida por la odometría.

La solución propuesta se muestra esquematizada en la figura 5.1. Se distinguen los siguientes puntos:

- En primer lugar, usando un método de detección y seguimiento de marcas naturales, se obtiene un conjunto de N trayectorias $\mathcal{T}^1, \dots, \mathcal{T}^N$ en el plano imagen. Cada una de las trayectorias $\mathcal{T}^i$, también denominadas “tracks” en la literatura, muestra la evolución temporal en el plano imagen de un punto perteneciente a la estructura en movimiento del robot.
- Una vez obtenido el conjunto de trayectorias, se obtiene la posición del robot mediante la odometría relativa a la posición inicial X_0 , aún desconocida. Desde un punto de vista estadístico se obtiene la distribución $p(X_1, \dots, X_K|X_0)$, dependiente funcionalmente del parámetro desconocido X_0 .
- Mediante un proceso iterativo de optimización, se obtiene el valor de X_0^a que maximiza la distribución $p(\mathcal{T}^1, \dots, \mathcal{T}^N|X_0^a)$. Dicha distribución se obtiene a partir de $p(X_1, \dots, X_K|X_0)$ y los modelos de formación de la imagen. Como se comentará en detalle en esta sección, maximizar la verosimilitud permitirá aprovechar de una manera eficiente la información dada por la odometría, teniendo en cuenta el error acumulativo que presenta.

A continuación se detalla la construcción de la función de verosimilitud y el algoritmo para obtener la solución.

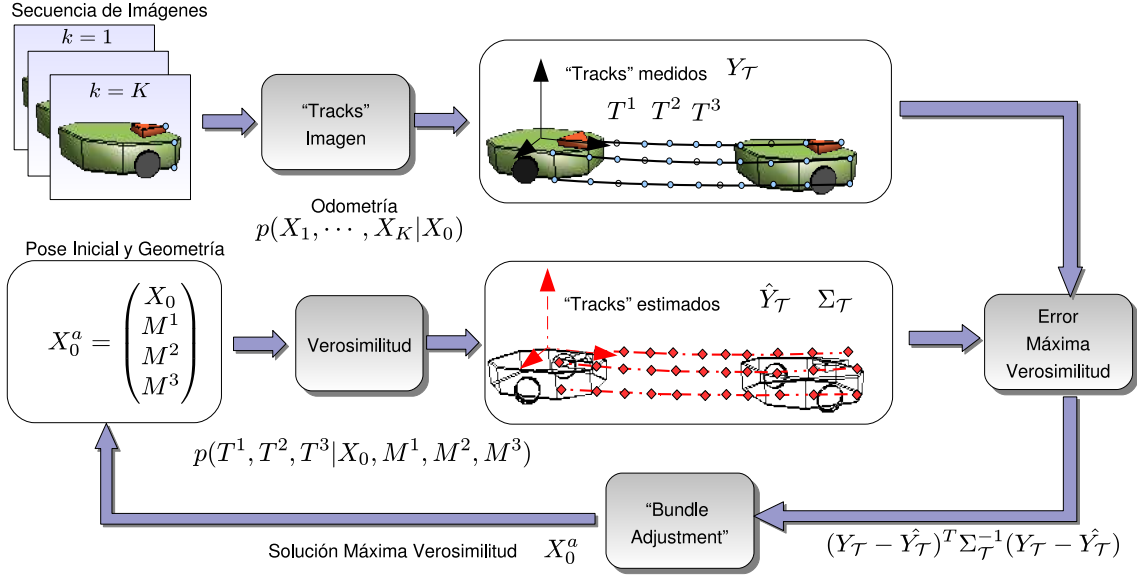


Figura 5.1: Inicialización de Máxima Verosimilitud, mediante la reducción de la discrepancia entre las medidas estimadas (círculos rojos) y las medidas tomadas en la imagen (círculos azules).

5.2.1. Odometría y medidas en el plano imagen

Las medidas de odometría $U_1, \dots, U_K$ se registran en una secuencia de K instantes de tiempo consecutivos $k = 1, \dots, K$ que corresponden a la trayectoria de inicialización realizada por el robot. Mediante el modelo de movimiento $g(X_{k-1}, U_k)$ descrito en la expresión (3.3), dada la posición inicial X_0 , se obtiene una estimación de la posición del robot relativa a dicha posición de la secuencia de inicialización completa:

$$\begin{aligned}
 X_1 &= g(X_0, U_1 + W_1) \\
 X_2 &= g(X_1, U_2 + W_2) \\
 &\vdots \\
 X_K &= g(X_{K-1}, U_K + W_K),
 \end{aligned} \tag{5.1}$$

donde W_k representa una realización del proceso $\mathbf{W}_k$ gaussiano de media nula, que modela el ruido aleatorio presente en las lecturas de odometría U_k .

Mediante el uso del modelo de movimiento, y al igual que se comentó en el capítulo 3, se modela la pose como un proceso aleatorio $\mathbf{X}_k$, condicionado en este caso a la pose inicial X_0 . De este modo, en cada instante de tiempo, se tiene la distribución de probabilidad conjunta $p(X_1, \dots, X_k | X_0)$ que se detalla a continuación:

$$\begin{aligned}
& p(X_1|X_0) \\
& p(X_1, X_2|X_0) = p(X_2|X_1) \cdot p(X_1|X_0) \\
& \vdots \\
& p(X_1, \dots, X_K|X_0) = p(X_K|X_{K-1}) \cdots p(X_{K-1}|X_0),
\end{aligned} \tag{5.2}$$

donde se entiende que la odometría U_k en cada instante de tiempo está incluida en la función de tránsito $p(X_k|X_{k-1})$ como fue descrito en el capítulo 3.

La distribución conjunta de las poses del robot, agrupadas en un solo vector columna X y condicional a la pose inicial X_0 en el intervalo $k = 1, \dots, K$ de inicialización será:

$$p(X|X_0) = \prod_{k=1}^K p(X_k|X_{k-1}) \quad X = (X_1^T \cdots X_K^T)^T \tag{5.3}$$

El equivalente gaussiano de (5.3), utilizando para ello una aproximación de primer orden, se presenta a continuación:

En primer lugar, se reagrupa la expresión (5.1) para convertirla en una única función vectorial que da como resultado el vector X dada la pose inicial X_0 y un vector con las componentes de la odometría incluyendo el ruido gaussiano en las mismas:

$$U = ((U_1 + W_1)^T \cdots (U_K + W_K)^T)^T \tag{5.4}$$

$$X = g_X(X_0, U) = \begin{pmatrix} g(X_0, U_1 + W_1) \\ g(g(X_0, U_1 + W_1), U_2 + W_2) \\ \vdots \\ g(g(\cdots (g(X_0, U_1 + W_1), U_2 + W_2), \cdots), U_K + W_K) \end{pmatrix} \tag{5.5}$$

El vector U se puede representar como un proceso gaussiano $\mathbf{U}$ de media $\hat{U}$, y matriz de covarianza Σ_U diagonal por bloques:

$$\hat{U} = (U_1^T \cdots U_K^T)^T \quad \Sigma_U = \begin{pmatrix} \Sigma_W & \cdots & \bar{0} \\ \vdots & \ddots & \vdots \\ \bar{0} & \cdots & \Sigma_W \end{pmatrix} \tag{5.6}$$

Mediante la aproximación de primer orden de g_X en torno a $\hat{U}$ se obtiene fácilmente la aproximación gaussiana de $p(X|X_0)$:

$$X \approx g_X(X_0, \hat{U}) + \frac{\partial g_X}{\partial U} W, \tag{5.7}$$

donde W representa un vector columna con las componentes de ruido presentes en U .

$$W = (W_1^T \quad \cdots \quad W_K^T) \quad (5.8)$$

Los valores de $\hat{X}$ y Σ_X , y por tanto la distribución de probabilidad que definen $p(X|X_0)$ se aproximan mediante el uso de la expresión linealizada (5.7):

$$\hat{X} = g_X(X_0, \hat{U}) \quad \Sigma_X = \frac{\partial g_X}{\partial U} \Sigma_U \frac{\partial g_X}{\partial U}^T \quad (5.9)$$

$$p(X|X_0) = \frac{1}{2^{K \cdot n_x/2} |\Sigma_X|} e^{-\frac{1}{2}(X - \hat{X})^T \Sigma_X^{-1} (X - \hat{X})} \quad (5.10)$$

Se demuestra fácilmente que Σ_X es, en general, una matriz de covarianza completa y por tanto todos los elementos del vector X son dependientes estadísticamente entre sí. De manera intuitiva, es fácil observar que debido a la naturaleza recursiva del modelo de movimiento, el ruido en la pose cometido en un instante de tiempo se propaga al siguiente, lo cual se traduce en dicha dependencia estadística.

Como se mostrará en este capítulo, el hecho de que Σ_X sea completa tendrá influencia en aspectos clave de la inicialización, como son la posibilidad de eliminar eficientemente los “outliers” de la medida, el tiempo de cómputo requerido o la influencia en el error de reconstrucción. Por este hecho, en esta sección se propondrán alternativas consistentes en aproximar la matriz Σ_X mediante una versión de la misma donde la correlación entre los distintos instantes de tiempos sea nula, dando lugar a una matriz Σ_X diagonal por bloques:

$$\Sigma_X \approx \begin{pmatrix} \Sigma_1 & \cdots & \bar{0} \\ \vdots & \ddots & \vdots \\ \bar{0} & \cdots & \Sigma_K \end{pmatrix} \quad (5.11)$$

El uso de la aproximación expuesta en 5.11 o de la matriz completa original será indicado en cada ocasión de manera explícita.

La distribución (5.3) representa la información métrica que posee el sistema acerca del movimiento realizado por el robot en la trayectoria usada en la inicialización. El objetivo planteado consiste en establecer una relación estadística entre la distribución (5.3), dependiente de la solución inicial X_0 para la pose, con los puntos pertenecientes a la estructura del robot $M^1, \dots, M^N$ y sus correspondientes medidas en el plano imagen. Dicha relación se obtendrá a continuación en forma de función de verosimilitud.

Durante la secuencia de inicialización, un proceso de “tracking” de marcas naturales, comentado en el capítulo 4, obtiene un conjunto de medidas que corresponden con la proyección en el plano imagen de cada uno de los puntos de la estructura del robot $M^1, \dots, M^N$ en cada instante de tiempo.

El algoritmo de detección y “tracking” utilizado, obtiene un primer conjunto de medidas Y_1 :

$$Y_1 = ((y_1^1)^T \quad \cdots \quad (y_1^N)^T)^T, \quad (5.12)$$

el cual contiene N puntos $y_1^i, i = 1, \dots, N$. Para cada uno de ellos se obtiene una trayectoria en el plano imagen nombrada como $\mathcal{T}^i$. Cada trayectoria contiene un conjunto de $K^i \leq K$ puntos que corresponden a instantes de tiempos donde el punto correspondiente M^i a la estructura del robot es visible:

$$\mathcal{T}^i = \begin{pmatrix} y_1^i \\ \vdots \\ y_{K^i}^i \end{pmatrix}, \quad (5.13)$$

El método de “tracking” usado no garantiza que en general los puntos contenidos en una trayectoria sean ni consecutivos ni siquiera que existan para todos los instantes de tiempo correspondientes a la secuencia de inicialización $k = 0, \dots, K$.

Ejemplo: si se tiene un punto y_1^1 detectado en el primer instante de tiempo $k = 1$ en una secuencia donde $K = 8$, una posible trayectoria visual $\mathcal{T}^1$ podría ser la mostrada en (5.14). En ella el proceso de “tracking” mantuvo la detección en instantes consecutivos hasta el instante $k = 3$, volviendo a ser detectado en $k = 7$ y desapareciendo definitivamente en $k = 8$.

$$\mathcal{T}^1 = \begin{pmatrix} y_1^1 \\ y_2^1 \\ y_3^1 \\ y_7^1 \end{pmatrix}, \quad (5.14)$$

Considerando cada medida y_k^i como la realización concreta del proceso aleatorio $\mathbf{y}_k^i$, se construye la f.d.p. condicionada del conjunto de trayectorias $Y_{\mathcal{T}} = (\mathcal{T}^1, \dots, \mathcal{T}^N)$, conocido el valor de la solución inicial X_0^a :

$$p(\mathcal{T}^1, \dots, \mathcal{T}^N | X_0^a) = p(Y_{\mathcal{T}} | X_0^a), \quad (5.15)$$

donde todas las trayectorias son empaquetadas por comodidad en un vector columna de medidas global $Y_{\mathcal{T}}$.

La distribución (5.15) representa el conocimiento estadístico que se tiene de las medidas en función de la solución buscada X_0^a . En dicha distribución se encuentran incluidos el modelo de proyección de la cámara y los parámetros de calibración de los que depende, los modelos de movimiento del robot y las lecturas de odometría, así como los procesos asociados al ruido de medida y movimiento $\mathbf{V}_k$ y $\mathbf{W}_k$.

La aproximación analítica de (5.15) como una f.d.p. gaussiana, a partir de los elementos que la componen, y su dependencia funcional con el vector X_0^a se expone a continuación.

Se parte de la distribución expuesta en (5.3) que describe la estimación de movimiento dado por la odometría. Puesto que el movimiento del robot no depende de los puntos de su estructura $M^1, \dots, M^N$, la siguiente igualdad es cierta:

$$p(X_1, \dots, X_K | X_0) = p(X_1, \dots, X_K | X_0, M^1, \dots, M^N) = p(X_1, \dots, X_K | X_0^a) \quad (5.16)$$

La relación estadística de la pose del robot con su geometría se obtiene mediante la propagación de la distribución (5.16) a través de la dependencia funcional existente entre las trayectorias en el plano imagen y el vector X_0^a .

Dado el conjunto de trayectorias $\mathcal{T}^1, \dots, \mathcal{T}^N$, se define la siguiente función vectorial:

$$Y_{\mathcal{T}} = \begin{pmatrix} \mathcal{T}^1 \\ \vdots \\ \mathcal{T}^N \end{pmatrix} = \begin{pmatrix} f_{\mathcal{T}}^1(X_1, \dots, X_K, M^1) \\ \vdots \\ f_{\mathcal{T}}^N(X_1, \dots, X_K, M^N) \end{pmatrix} + V \quad V = \begin{pmatrix} V^1 \\ \vdots \\ V^N \end{pmatrix}, \quad (5.17)$$

donde V representa una realización concreta del proceso $\mathbf{V} = N(\bar{0}, \Sigma_V)$, que aglutina la incertidumbre en la medida del conjunto de trayectorias ya comentado en el capítulo 3:

$$\Sigma_{V^i} = \begin{pmatrix} \Sigma_v & \cdots & \bar{0} \\ \vdots & \ddots & \vdots \\ \bar{0} & \cdots & \Sigma_v \end{pmatrix} \rightarrow \Sigma_V = \begin{pmatrix} \Sigma_{V^1} & \cdots & \bar{0} \\ \vdots & \ddots & \vdots \\ \bar{0} & \cdots & \Sigma_{V^N} \end{pmatrix} \quad (5.18)$$

Cada una de las funciones $f_{\mathcal{T}}^i(X_1, \dots, X_K, M^i)$ representa la proyección en el plano imagen del punto M^i :

$$f_{\mathcal{T}}^i(X_1, \dots, X_K, M^i) = \begin{pmatrix} h(t(X_1, M^i), P) \\ \vdots \\ h(t(X_K, M^i), P) \end{pmatrix}, \quad (5.19)$$

Tanto (5.16) como su propagación a través de la expresión (5.17), para obtener la distribución buscada $p(Y_{\mathcal{T}} | X_0^a)$, resultan ser en la práctica funciones no analíticas sobre los parámetros que las definen.

Mediante el uso de técnicas de aproximación gaussiana, como puede ser la linealización de primer orden o la transformada “Unscented”, se obtiene con facilidad una aproximación gaussiana de $p(Y_{\mathcal{T}} | X_0^a)$, parametrizada por un vector media $\hat{Y}_{\mathcal{T}}$ y una matriz de covarianza $\Sigma_{\mathcal{T}}$:

$$p(Y_{\mathcal{T}} | X_0^a) \approx \frac{1}{2^{N_Y} |\Sigma_{\mathcal{T}}|} e^{-\frac{1}{2} (Y_{\mathcal{T}} - \hat{Y}_{\mathcal{T}})^T \Sigma_{\mathcal{T}}^{-1} (Y_{\mathcal{T}} - \hat{Y}_{\mathcal{T}})} \quad \Sigma_{\mathcal{T}} \in \mathbb{R}^{2N_Y \times 2N_Y}, \quad (5.20)$$

donde N_Y representa el número de puntos totales contenidos en el vector $Y_{\mathcal{T}}$.

Tanto $\hat{Y}_{\mathcal{T}}$ como $\Sigma_{\mathcal{T}}$ son en general funciones no lineales con respecto a la solución buscada X_0^a debido principalmente a los modelos de proyección de la imagen y movimiento del robot.

Se describe en el siguiente punto el uso de la distribución (5.20), utilizando para ello un enfoque de máxima verosimilitud, como base para la obtención de X_0^a que da solución al problema de la inicialización.

5.2.2. Enfoque de máxima verosimilitud

Se plantea como algoritmo de reconstrucción del vector X_0^a un criterio de Máxima Verosimilitud, consistente en encontrar el valor de X_0^a que maximiza la aproximación gaussiana de la f.d.p. $p(Y_{\mathcal{T}}|X_0^a)$:

$$\max_{X_0^a} p(Y_{\mathcal{T}}|X_0^a) = \max_{X_0^a} \left\{ \frac{1}{2^{N_Y} |\Sigma_{\mathcal{T}}|} e^{-\frac{1}{2}(Y_{\mathcal{T}} - \hat{Y}_{\mathcal{T}})^T \Sigma_{\mathcal{T}}^{-1} (Y_{\mathcal{T}} - \hat{Y}_{\mathcal{T}})} \right\}, \quad (5.21)$$

donde $Y_{\mathcal{T}}$ representa una realización concreta del proceso $\mathbf{Y}_{\mathcal{T}}$ determinado por la f.d.p. $p(Y_{\mathcal{T}}|X_0^a)$, es decir, las medidas concretas tomadas en el plano imagen para la inicialización.

Tomando el logaritmo de la expresión (5.21) y cambiando su signo, se obtiene un problema equivalente en el que se busca el mínimo en función de X_0^a de la siguiente función de coste:

$$\min_{X_0, M^1, \dots, M^N} (Y_{\mathcal{T}} - \hat{Y}_{\mathcal{T}})^T \Sigma_{\mathcal{T}}^{-1} (Y_{\mathcal{T}} - \hat{Y}_{\mathcal{T}}), \quad (5.22)$$

la cual se ha simplificado a una función de coste cuadrática, muy similar a la propuesta por los métodos de “Bundle Adjustment”, donde se utiliza una matriz de ponderación W en lugar de $\Sigma_{\mathcal{T}}^{-1}$. A diferencia de otras propuestas [Triggs et al., 2000], donde no existe una estimación de movimiento y el criterio a minimizar se basa en la distancia de reproyección, en este caso la matriz $\Sigma_{\mathcal{T}}$ es una aproximación de la covarianza real del proceso de medida y, por tanto, su uso supone una ventaja en la precisión obtenida para la estimación de X_0^a .

La estructura de la matriz $\Sigma_{\mathcal{T}}$ será completa, presentando términos de correlación cruzada no nulos tanto entre diferentes medidas como la misma medida en diferentes instantes de tiempo. Si Σ_X es aproximada por una matriz diagonal por bloques, entonces $\Sigma_{\mathcal{T}}$ presentará correlación nula en diferentes instantes de tiempo. De hecho, tras una reordenación del vector $Y_{\mathcal{T}}$, presentará una estructura diagonal por bloques. Una demostración ilustrativa de este hecho se consigue acudiendo a una aproximación de primer orden para la obtención de $\Sigma_{\mathcal{T}}$.

Se obtiene la aproximación de primer orden de la función (5.19) con respecto al conjunto de posiciones X :

$$f_{\mathcal{T}}^i(X + \Delta_X, M^i) \approx f_{\mathcal{T}}^i(X, M^i) + J_X^i \Delta_X, \quad (5.23)$$

donde:

$$J_X^i = \begin{pmatrix} \frac{\partial y_1^i}{\partial X_1} & \dots & \bar{0} \\ \vdots & \ddots & \vdots \\ \bar{0} & \dots & \frac{\partial y_K^i}{\partial X_K} \end{pmatrix}, \quad (5.24)$$

siendo:

$$\frac{\partial y_k^i}{\partial X_k} = \frac{\partial h(t(X_k, M^i), P)}{\partial X_k} \quad (5.25)$$

Extendiendo la aproximación de primer orden a la expresión (5.17) se obtiene:

$$Y_{\mathcal{T}}(X + \Delta_X) = \begin{pmatrix} f_{\mathcal{T}}^1(X_0, \dots, X_K, M^1) \\ \vdots \\ f_{\mathcal{T}}^N(X_0, \dots, X_K, M^N) \end{pmatrix} + J_X \Delta_X + V, \quad (5.26)$$

siendo J_X :

$$J_X = \begin{pmatrix} J_X^1 \\ \vdots \\ J_X^N \end{pmatrix} \quad (5.27)$$

Utilizando la aproximación de primer orden, y suponiendo independencia estadística entre $\mathbf{X}$ y $\mathbf{V}$ se obtiene la media $\hat{Y}_{\mathcal{T}}$ y la varianza $\Sigma_{\mathcal{T}}$:

$$\hat{Y}_{\mathcal{T}} = \begin{pmatrix} f_{\mathcal{T}}^1(\hat{X}, M^1) \\ \vdots \\ f_{\mathcal{T}}^N(\hat{X}, M^N) \end{pmatrix} \quad \Sigma_{\mathcal{T}} = J_X \Sigma_X J_X^T + \Sigma_V \quad (5.28)$$

En la figura 5.2 se observa como la estructura de la matriz J_X transforma la matriz diagonal por bloques Σ_X en una matriz donde los términos de $Y_{\mathcal{T}}$ pertenecientes a diferentes instantes de tiempo son independientes entre sí. Para analizar este hecho con mas claridad se realiza la siguiente ordenación del vector $Y_{\mathcal{T}}$ que da lugar al vector Y_L , en el cual medidas tomadas en un instante de tiempo están agrupadas en un solo vector Y_k :

$$Y_{\mathcal{T}} = \begin{pmatrix} y_1^1 \\ \vdots \\ y_K^1 \\ y_1^2 \\ \vdots \\ y_K^2 \\ \vdots \\ y_K^N \end{pmatrix} \Rightarrow Y_L = \begin{pmatrix} y_1^1 \\ \vdots \\ y_1^N \\ y_2^1 \\ \vdots \\ y_2^N \\ \vdots \\ y_K^N \end{pmatrix} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_K \end{pmatrix} \quad (5.29)$$

En ese caso, denominando Σ_L a la matriz de covarianza de Y_L , se ilustra mas claramente la independencia entre diferentes instantes de tiempo que presenta la matriz de covarianza $\Sigma_{\mathcal{T}}$ y que se puede observar en la estructura diagonal por bloques de Σ_L (ver figura 5.2). Si se denomina a cada bloque independiente como Σ_{Y_k} , donde k indica el instante de tiempo a que pertenece, la matriz Σ_L se expresa del siguiente modo:

$$\Sigma_L = \begin{pmatrix} \Sigma_{Y_1} & \cdots & \bar{0} \\ \vdots & \ddots & \vdots \\ \bar{0} & \cdots & \Sigma_{Y_K} \end{pmatrix} \quad \Sigma_{Y_k} \in \mathbb{R}^{n_y \times N} \quad (5.30)$$

El hecho de que las medidas en el plano imagen estén correladas entre si para cada instante de tiempo será importante en aspectos clave del algoritmo de inicialización.

$$\underbrace{\left(\begin{matrix} \text{[Sparse blocks]} \\ J_X \end{matrix} \right)}_{J_X} \circ \underbrace{\left(\begin{matrix} \text{[Small blocks]} \\ \Sigma_X \end{matrix} \right)}_{\Sigma_X} \circ \underbrace{\left(\begin{matrix} \text{[Small blocks]} \\ J_X^T \end{matrix} \right)}_{J_X^T} + \underbrace{\left(\begin{matrix} \text{[Diagonal blocks]} \\ \Sigma_V \end{matrix} \right)}_{\Sigma_V} = \underbrace{\left(\begin{matrix} \text{[Large sparse blocks]} \\ \Sigma_T \end{matrix} \right)}_{\Sigma_T} \Rightarrow \underbrace{\left(\begin{matrix} \text{[Large block-diagonal]} \\ \Sigma_L \end{matrix} \right)}_{\Sigma_L}$$

Figura 5.2: Estructura de la matriz Σ_T , formada a partir de la transformación de la matriz Σ_X mediante el proceso de medida linealizado. Cada cuadrado sombreado representa un valor no nulo en la matriz. El tamaño de las matrices se ha dispuesto, a modo aclaratorio, para un total de $K = 3$ instantes de tiempo y un total de $N = 4$ puntos del modelo.

La estructura de la matriz de covarianza tendrá efectos sustanciales en varios aspectos destacados de la solución propuesta, los cuales, se detallarán a lo largo de esta sección. Por simplicidad, a partir de este punto se utilizará el proceso $\mathbf{Y}_L$, resultado de la reordenación de $\mathbf{Y}_T$.

Como ha sido descrito con anterioridad el objetivo es la determinación del vector X_0^a que minimiza la expresión (5.22). Las diferentes aproximaciones de la matriz Σ_L transforman la función de coste en las siguientes posibilidades, cuyas propiedades serán estudiadas en este capítulo:

1. **Dependencia estadística completa:** En este caso, la matriz Σ_L no tiene una estructura interna que permita subdividirla en bloques y por tanto, todas las dependencias estadísticas están contempladas. La función de coste resultante será la misma que (5.22), salvo que por conveniencia se utiliza el vector reordenado Y_L :

$$\min_{X_0^a} (Y_L - \hat{Y}_L)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L), \quad (5.31)$$

2. **Independencia estadística temporal:** La aproximación de Σ_L , basada en considerar que la estimación de movimiento no está correlada entre diferentes instantes de tiempo, resulta en la siguiente función de coste:

$$\min_{X_0^a} \sum_{k=1}^K (Y_k - \hat{Y}_k)^T \Sigma_{Y_k}^{-1} (Y_k - \hat{Y}_k), \quad (5.32)$$

donde cada submatriz Σ_{Y_k} corresponde al bloque que forma la versión de Σ_L diagonal por bloques.

3. **Independencia estadística temporal y estructural:** Se introduce en este punto una nueva aproximación de Σ_L que en concreto elimina las dependencias estadísticas entre diferentes marcas y entre diferentes instantes de tiempo. En este caso, la matriz resultante será diagonal en bloques $\Sigma_{y_k^i}$ de dimensión 2×2 , los cuales, describen la dependencia estadística entre los diferentes elementos del vector y_k^i (es decir, ambas coordenadas de la imagen). En este caso concreto la nueva función de coste será la siguiente:

$$\min_{X_0^a} \sum_{k=1}^K \sum_{i=1}^N (y_k^i - \hat{y}_k^i)^T \Sigma_{y_k^i}^{-1} (y_k^i - \hat{y}_k^i), \quad (5.33)$$

4. **Ausencia de modelado estadístico:** Es interesante notar que si se obvia el modelado estadístico propuesto, suponiendo que Σ_L corresponde con una matriz identidad, la función de coste resultante será la suma de los errores residuales en cada instante de tiempo y para cada una de las marcas presentes. Dicha función de coste es utilizada con frecuencia en los procesos de reconstrucción o de “Bundle Adjustment”, y es por ello que resulta como una opción de comparación con respecto al criterio de máxima verosimilitud:

$$\min_{X_0^a} \sum_{k=1}^K \sum_{i=1}^N (y_k^i - \hat{y}_k^i)^T (y_k^i - \hat{y}_k^i), \quad (5.34)$$

El mínimo de cualquiera de las funciones de coste planteadas (5.31) (5.32) (5.33) (5.34) se obtiene mediante la utilización de un algoritmo iterativo de optimización. En esta tesis se hace uso del algoritmo Levenberg-Marquardt, el cual es usual en procesos de reconstrucción de la misma índole.

El método de Levenberg-Marquardt es esencialmente un algoritmo de optimización de tipo Gauss-Newton, en el cual se aproxima la función de coste por una función cuadrática en un entorno de X_0^a . De este modo, las dos primeras derivadas (Jacobiano y Hessiano) de la función de coste con respecto a los parámetros deben estar disponibles.

Los métodos de Gauss-Newton, requieren que localmente al mínimo que se busca la función sea lo más próxima posible a una función cuadrática. En el caso de las funciones de coste propuestas, el principal problema que presentan para ser minimizadas por un criterio Gauss-Newton es la dependencia funcional de la varianza Σ_L con respecto a los parámetros

X_0^a . Dicha dependencia es explícita si se utilizan linealizaciones para la obtención de Σ_L , como ha sido expuesto en la expresión (5.28).

Este hecho tiene una repercusión clara en el algoritmo de minimización, puesto que afecta a la suposición cuadrática, incrementando a su vez el número de mínimos locales de la función, donde el algoritmo puede quedar atrapado. En esta tesis se propone aproximar las derivadas suponiendo que la matriz Σ_L no es función de X_0^a , aunque actualizando el valor de Σ_L en cada iteración. Esta aproximación permite reducir el incremento en la no linealidad de la función de coste a expensas de decrementar la precisión en la actualización de la solución en cada iteración. Una breve descripción del método de Levenberg-Marquardt utilizado se presenta en el algoritmo 5, donde se obtiene el mínimo de la función de coste general (5.31).

Algoritmo 5 Obtención del mínimo de (5.31) mediante Levenberg-Marquardt

- 1: **Requerimientos:**
- 2: Medidas en el plano imagen Y_L
- 3: Lecturas de odometría $U_1, \dots, U_K$
- 4: Parámetros para los modelos de movimiento g y proyección h
- 5: Valor inicial para X_0^a , $numiter_{max}$ y ϵ_{min}
- 6:
- 7: **while** $\epsilon > \epsilon_{min}$ **and** $numiter < numiter_{max}$ **do**
- 8: Obtener $\hat{Y}_L$ y Σ_L usando (5.28).
- 9: Cálculo de J y H .

$$J = -2\left(\frac{\partial \hat{Y}_L}{\partial X_0^a}\right)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L) \quad (5.35)$$

$$H \approx 2\left(\frac{\partial \hat{Y}_L}{\partial X_0^a}\right)^T \Sigma_L^{-1} \left(\frac{\partial \hat{Y}_L}{\partial X_0^a}\right) \quad (5.36)$$

- 10: $\epsilon_{LM} = \infty$
 - 11: **while** $\epsilon_{LM} > \epsilon$ **do**
 - 12: $H_{LM} = H + \delta I$
 - 13: $\Delta_{X^a} = -(H)^{-1} J$
 - 14: $X_{LM}^a = X_0^a + \Delta_{X^a}$
 - 15: Obtener $\hat{Y}_L$ y Σ_L usando (5.28) y el valor X_{LM}^a
 - 16: $\epsilon_{LM} = (Y_L - \hat{Y}_L)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L)$
 - 17: $\delta = \delta \cdot 10$
 - 18: **end while**
 - 19: $\epsilon = \epsilon_{LM}$
 - 20: $\delta = \delta/100$ $X_0^a = X_{LM}^a$
 - 21: **end while**
-

5.2.3. Influencia del modelado estadístico en la solución

Cada una de las estrategias que se han comentado, que dan lugar a las funciones de coste planteadas, tendrán un impacto directo en la solución de la pose y geometría del robot. La influencia se puede clasificar en los siguientes factores que serán desglosados y comentados en éste capítulo:

- **Error de reconstrucción:** Se entiende por error de reconstrucción a la diferencia

entre el vector X_0^a estimado, con respecto a la solución real o “Ground Truth”. Como se verá más adelante, el hecho de introducir la ponderación dada por Σ_L en la función de coste permitirá obtener un error de reconstrucción menor que en el caso de no introducir conocimiento estadístico.

- **Tiempo de Cómputo:** Como se ha visto en el algoritmo 5, en el caso de utilizar la matriz Σ_L , ésta requiere su recómputo en cada iteración del algoritmo de optimización. Sin embargo, el uso de Σ_L tiene a su vez un impacto directo en las propiedades matriciales en el Hessiano H , el cual, en caso de no usar modelos estadísticos, tiende a ser una matriz “dispersa” y por tanto la complejidad algorítmica para obtener su inversa en cada iteración disminuye. Este hecho puede ser de suma importancia en el caso de que X_0^a contenga un gran número de puntos, es decir un valor para N grande.
- **Tratamiento de los “outliers” en la medida:** El algoritmo 5 no tiene en cuenta que algunos elementos del vector Y_L puedan ser producto de una mala detección en el plano imagen, o que incluso pertenezcan a otros objetos en movimiento. En este caso existen una serie de modificaciones posibles en la función de coste que permiten modelar la aparición de dichos puntos, denominados “outliers” y que minimizan su aportación errónea a la solución final. La influencia de las diferentes aproximaciones de Σ_L , y su impacto en el diseño de algoritmos robustos serán convenientemente detalladas en este capítulo.

5.2.4. Error de reconstrucción

Para ilustrar la influencia que tienen las distintas versiones de Σ_L , incluyendo el caso de suponerla una matriz identidad, en la solución final de X_0^a , se propone un experimento sencillo que comparta las propiedades esenciales con el problema planteado y que se presenta a continuación.

Se plantea reducir la dimensionalidad del problema propuesto, en el cual, mediante una cámara proyectiva se obtiene una imagen a partir de puntos tridimensionales del robot. En su lugar se utilizará una analogía más sencilla en la que se tiene un mundo bidimensional proyectado en una cámara proyectiva unidimensional. El objetivo principal es reducir la dimensión de las matrices involucradas y facilitar los ejemplos gráficos del problema.

Se propone un modelo de robot compuesto por un conjunto de tres puntos $M = \{M^1, \dots, M^3\}$ definidos en un plano, los cuales rotan alrededor del punto $(0,0)^T$ que define el sistema de coordenadas del robot. Las coordenadas en el plano de los tres puntos con respecto al origen de rotación son las siguientes:

$$\begin{aligned} M^1 &= (-l/2, l/2)^T \\ M^2 &= (-l/2, -l/2)^T \\ M^3 &= (l, 0)^T \end{aligned} \quad l = 20(mm), \quad (5.37)$$

en las que se han establecido unidades métricas para el parámetro l con el objeto de presentar resultados numéricos que permitan ilustrar el ejemplo. La posición y orientación de los tres puntos con respecto a un origen de coordenadas global, en el instante k se

determina con un vector de 3 componentes $X_k = (x_k^1 \ x_k^2 \ \alpha_k)$. El vector aumentado análogo al introducido para describir la pose y estructura de un robot será por tanto:

$$X_k^a = \begin{pmatrix} X_k \\ M^1 \\ M^2 \\ M^3 \end{pmatrix} \quad (5.38)$$

El modelo de movimiento g en este ejemplo es igual que el utilizado para un robot diferencial y comentado en (3.3). De este modo, partiendo de una posición inicial X_0 y dadas unas lecturas de odometría contaminadas con ruido $\mathbf{W}_k = N(O, \Sigma_W)$, se obtiene una estimación de la posición del objeto.

Se propone el uso de una cámara proyectiva unidimensional como la mostrada en la figura 5.3. Para simplificar se supone que la línea de proyección (equivalente al plano de proyección) es paralela al eje vertical del plano (coordenada x^2) y la cámara se encuentra posicionada en el origen de coordenadas del plano.

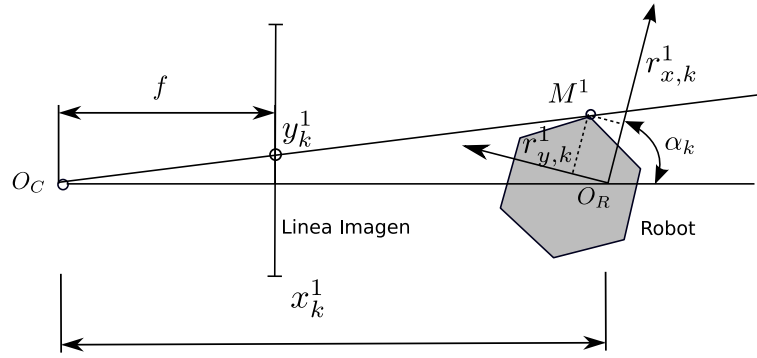


Figura 5.3: Cámara proyectiva unidimensional que proyecta un mundo bidimensional

La proyección en el plano imagen del punto $M^1 = (r_x^1, r_y^1)^T$ en el instante k será:

$$M_k^1 = \begin{pmatrix} r_{xk}^1 \\ r_{yk}^1 \end{pmatrix} = \begin{pmatrix} r_{xk} \\ r_{yk} \end{pmatrix} + \begin{pmatrix} \cos(\alpha_k) & -\sin(\alpha_k) \\ \sin(\alpha_k) & \cos(\alpha_k) \end{pmatrix} M^1 \quad (5.39)$$

$$y_k^1 = f \frac{r_{yk}^1}{r_{xk}^1} + v_k^1 \quad \mathbf{v}_k = N(0, \sigma_v^2) \quad f = 40mm \quad (5.40)$$

Para que el ejemplo sea ilustrativo, los procesos de ruido $\mathbf{W}_k$ y $\mathbf{V}_k$ tienen las mismas propiedades estadísticas ya comentadas con anterioridad.

El problema de la inicialización, utilizando las condiciones expuestas en este ejemplo, tiene idéntico planteamiento con el problema real de esta tesis, salvo el hecho de que la estructura del robot es plana y la cámara es de proyección unidimensional.

Con este ejemplo se pretende ilustrar la influencia que existe al introducir las distintas aproximaciones de Σ_L en la función de coste, cuyo mínimo tiene como resultado el vector

X_0^a . Para ello, en lugar de implementar un algoritmo de optimización para el problema reducido que se propone, se discutirá el coste de error que tiene una solución intencionadamente errónea $X_0^a + \Delta_{X^a}$ en las distintas alternativas para Σ_L , comparado con la solución exacta. Se mostrará cómo en determinadas configuraciones, la solución errónea presenta menor error medido por la función de coste que aquella correcta, lo que justifica el hecho de utilizar el criterio de máxima verosimilitud para evitar este hecho.

En la figura 5.4 se muestra una secuencia con $K = 3$ posiciones del robot “triangular”, generadas por los vectores de odometría $U_{1,2,3} = (50 \text{ mm/s} \ 15^\circ/\text{s})^T$ y una pose inicial $X_0 = (70 \text{ mm} \ -100 \text{ mm} \ 100^\circ)$. La matriz de covarianza de la odometría es $\Sigma_W = \text{diag}(20, 2)$. La longitud focal de la cámara $f = 40 \text{ mm}$ y el error unidimensional de medida es de $\Sigma_v = 2 \text{ mm}^2$. La “línea imagen” de la cámara se representa como una línea vertical en la que intersectan los rayos que unen la geometría del robot, rotada y trasladada mediante X_k , y el origen de coordenadas de la cámara.

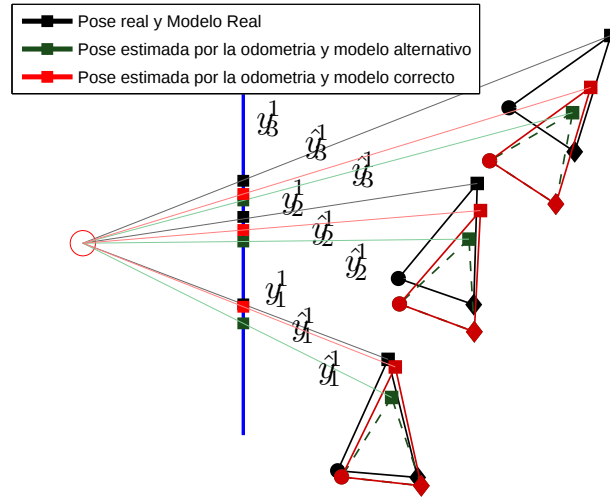


Figura 5.4: Ejemplo ilustrativo de una secuencia de $K = 3$ posiciones del robot y su proyección en la cámara lineal mediante rayos que conectan un foco con los puntos del robot. Se muestran las posiciones reales y aquellas estimadas por la odometría para dos modelos, uno correcto y otro intencionadamente erróneo (línea punteada)

El vector solución buscado X_0^a será el siguiente:

$$X_0^a = \left(\underbrace{\begin{pmatrix} 70 & -100 & 100 \end{pmatrix}}_{X_0^T} \underbrace{\begin{pmatrix} -10 & 10 \end{pmatrix}}_{(M^1)^T} \underbrace{\begin{pmatrix} -10 & -10 \end{pmatrix}}_{(M^2)^T} \underbrace{\begin{pmatrix} 20 & 0 \end{pmatrix}}_{(M^3)^T} \right)^T \quad (5.41)$$

La matriz de covarianza completa para la posición del robot Σ_X se calcula utilizando una versión simplificada de la expresión (5.9):

$$\Sigma_X = \begin{pmatrix} \boxed{\begin{matrix} 2,76 & -4,62 & -1,69 \\ -4,62 & 18,76 & -0,45 \\ -1,69 & -0,45 & 2,00 \end{matrix}} & \begin{matrix} 3,58 & -3,40 & -1,69 \\ -4,40 & 19,09 & -0,45 \\ -2,66 & -1,90 & 2,00 \end{matrix} & \begin{matrix} 2,84 & -2,13 & -1,69 \\ -4,60 & 19,43 & -0,45 \\ -1,78 & -3,41 & 2,00 \end{matrix} \\ \begin{matrix} 3,58 & -4,40 & -2,66 \\ -3,40 & 19,09 & -1,90 \\ -1,69 & -0,45 & 2,00 \end{matrix} & \boxed{\begin{matrix} 8,99 & -8,89 & -4,28 \\ -8,89 & 37,87 & -2,55 \\ -4,28 & -2,55 & 4,00 \end{matrix}} & \begin{matrix} 7,11 & -5,67 & -4,28 \\ -10,02 & 39,79 & -2,55 \\ -2,51 & -5,57 & 4,00 \end{matrix} \\ \begin{matrix} 2,84 & -4,60 & -1,78 \\ -2,13 & 19,43 & -3,41 \\ -1,69 & -0,45 & 2,00 \end{matrix} & \begin{matrix} 7,11 & -10,02 & -2,51 \\ -5,67 & 39,79 & -5,57 \\ -4,28 & -2,55 & 4,00 \end{matrix} & \boxed{\begin{matrix} 12,14 & -16,13 & -4,03 \\ -16,13 & 59,37 & -6,44 \\ -4,03 & -6,44 & 6,00 \end{matrix}} \end{pmatrix} \quad (5.42)$$

En una secuencia con $K = 3$ existen medidas en el intervalo $k = 1, 2, 3$, que forman el vector Y_L formado por 9 componentes:

$$Y_L = ((y_1^1)^T \quad (y_1^2)^T \quad (y_1^3)^T \quad (y_2^1)^T \quad (y_2^2)^T \quad (y_2^3)^T \quad (y_3^1)^T \quad (y_3^2)^T \quad (y_3^3)^T)^T \quad (5.43)$$

Se distinguirán tres vectores de medida:

- El vector Y_L corresponde a las proyecciones en la cámara lineal de la estructura del robot, usando para ello las lecturas de odometría libres de error (color negro en la figura 5.4) y el vector X_0^a . Cada componente y_k^i está sujeta a un ruido aditivo de media nula representado por Σ_V .
- El vector $\hat{Y}_L$, correspondería a la estimación realizada por el sistema mediante las lecturas de odometría con un ruido aleatorio descrito por Σ_W y el valor del vector X_0^a (color rojo en la figura 5.4).
- Por último, el vector de medidas $\hat{Y}_L^\epsilon$ corresponde al conjunto de medidas estimadas en la “línea imagen” mediante el uso de una solución inicial $X_0^a + \Delta_{X^a}$ errónea, y el vector de odometría contaminado con ruido (color verde en la figura 5.4).

La modificación del vector de estados, descrito por Δ_{X^a} , consiste en un simple desplazamiento en un punto concreto de la estructura del robot:

$$X_0^a + \Delta_{X^a} = \left((X_0)^t \quad (M^1)^T \quad (M^2)^T \quad (M^3)^T + \begin{pmatrix} 8 & 0 \end{pmatrix}^T \right)^T. \quad (5.44)$$

En la figura 5.4 se observan las posiciones en la “línea imagen” de los vectores $\hat{Y}_L$ y $\hat{Y}_L^\epsilon$, comparados ambos con las medidas reales Y_L .

Mediante el uso de la aproximación de primer orden, mostrada en (5.28), se calcula la matriz Σ_L del proceso $\hat{Y}_L$ y la correspondiente Σ_L^ϵ del proceso $\hat{Y}_L^\epsilon$.

$$\Sigma_L = \begin{pmatrix} \boxed{4,40} & 2,15 & 2,49 & 3,00 & 2,66 & 3,13 & 3,98 & 3,48 & 4,34 \\ 2,15 & \boxed{3,96} & 2,24 & 2,63 & 2,37 & 2,78 & 3,45 & 3,06 & 3,82 \\ 2,49 & 2,24 & \boxed{4,59} & 3,11 & 2,76 & 3,25 & 4,12 & 3,61 & 4,50 \\ 3,00 & 2,63 & 3,11 & \boxed{8,94} & 6,12 & 7,32 & 9,52 & 8,28 & 10,45 \\ 2,66 & 2,37 & 2,76 & 6,12 & \boxed{7,47} & 6,51 & 8,26 & 7,28 & 9,18 \\ 3,13 & 2,78 & 3,25 & 7,32 & 6,51 & \boxed{9,76} & 9,95 & 8,73 & 11,01 \\ 3,98 & 3,45 & 4,12 & 9,52 & 8,26 & 9,95 & \boxed{20,56} & 16,12 & 20,62 \\ 3,48 & 3,06 & 3,61 & 8,28 & 7,28 & 8,73 & 16,12 & \boxed{16,16} & 18,09 \\ 4,34 & 3,82 & 4,50 & 10,45 & 9,18 & 11,01 & 20,62 & 18,09 & \boxed{25,13} \end{pmatrix} \quad (5.45)$$

$$\Sigma_L^\epsilon = \begin{pmatrix} \boxed{4,40} & 2,15 & 2,43 & 3,00 & 2,66 & 3,05 & 3,98 & 3,48 & 4,16 \\ 2,15 & \boxed{3,96} & 2,19 & 2,63 & 2,37 & 2,70 & 3,45 & 3,06 & 3,65 \\ 2,43 & 2,19 & \boxed{4,47} & 3,03 & 2,69 & 3,08 & 4,01 & 3,52 & 4,20 \\ 3,00 & 2,63 & 3,03 & \boxed{8,94} & 6,12 & 7,10 & 9,52 & 8,28 & 10,00 \\ 2,66 & 2,37 & 2,69 & 6,12 & \boxed{7,47} & 6,31 & 8,26 & 7,28 & 8,76 \\ 3,05 & 2,70 & 3,08 & 7,10 & 6,31 & \boxed{9,29} & 9,66 & 8,46 & 10,20 \\ 3,98 & 3,45 & 4,01 & 9,52 & 8,26 & 9,66 & \boxed{20,56} & 16,12 & 19,66 \\ 3,48 & 3,06 & 3,52 & 8,28 & 7,28 & 8,46 & 16,12 & \boxed{16,16} & 17,22 \\ 4,16 & 3,65 & 4,20 & 10,00 & 8,76 & 10,20 & 19,66 & 17,22 & \boxed{22,96} \end{pmatrix}$$

	Solución correcta X_0^a	Solución alterada $X_k^a + \Delta_{X^a}$
Matriz completa (M.C.)	$\epsilon = 8,8530$	$\epsilon = 17,8131$
Aproximación por bloques (M.B.)	$\epsilon = 7,5000$	$\epsilon = 15,3317$
Aproximación diagonal (M. D.)	$\epsilon = 7,8763$	$\epsilon = 9,4938$
Matriz identidad (M.I.)	$\epsilon = 119,8060$	$\epsilon = 87,4594$

Tabla 5.1: Comparación del error de coste obtenido con las diferentes alternativas para la matriz Σ_L , tanto para una solución correcta como para una solución alterada.

En la tabla 5.1 se muestra el residuo ϵ resultante de aplicar cada una de las aproximaciones de la función de coste. Es decir, para un caso general:

$$\epsilon = (Y_L - \hat{Y}_L)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L) \quad (5.46)$$

Como se observa en la mencionada tabla, las opciones existentes donde existe un modelado estadístico (M.C., M.B y M.D.), el error resultante de cada función de coste correspondiente es inferior cuando se trata de la solución correcta X_0^a , con respecto a la que se encuentra alterada $X_0^a + \Delta_{X^a}$. Sin embargo, utilizando simplemente distancias euclídeas en la “línea imagen”, lo que correspondería a la matriz Σ_L identidad, revela por el contrario un error menor en el caso de la solución alterada. Se puede asegurar, por tanto, que el mínimo global de la función de esta última función de coste, (i.e. ausencia de modelado estadístico) no puede ser X_0^a ya que existe un valor alterado $X_0^a + \Delta_{X^a}$ que posee un valor de ϵ menor.

De este modo, el algoritmo de optimización con toda probabilidad no dará con la solución correcta utilizando una matriz Σ_L correspondiente a la matriz identidad.

Sin embargo, la pregunta esencial consiste en aseverar si alguna de las funciones de coste planteadas, que sí parecen acertar en otorgar un error menor a la solución correcta, tienen como mínimo global a X_0^a . Para ello se debería demostrar que no existe un valor para Δ_{X^a} tal que dé como resultado un error en la función de coste menor que el resultante de X_0^a .

Dicha aseveración no puede ser abordada mediante un método analítico sencillo, dada la no linealidad de la función de proyección y los modelos de movimiento del robot. En el caso del problema real con un mundo tridimensional, el cual es el objetivo de esta tesis, la problemática es considerablemente mayor y la respuesta a la pregunta carece de igual modo de justificación analítica. En la presente tesis se abordará esta cuestión, y resultados de similar complejidad, mediante experimentación en condiciones de contorno reales, mostrando así y en ese caso generalista las propiedades obtenidas en la reconstrucción de X_0^a en cada una de las opciones planteadas para la función de coste.

Sin embargo, para el problema reducido que se muestra en este apartado de la tesis, se pueden no obstante resaltar una serie de observaciones que justifican de manera aproximada las propiedades de un correcto modelado estadístico para componer la función de coste. Para ello se plantea proponer ejemplos parecidos al utilizado en la figura 5.4, que permitan mostrar las carencias e inmunidad de las distintas aproximaciones de Σ_L .

En primer lugar se resumen propiedades significativas del problema:

- El error obtenido en la estimación de movimiento es acumulativo y por tanto crece con la longitud de la trayectoria, siendo las últimas posiciones de la trayectoria aquellas con un error esperado mayor (véase Σ_X en (5.42)).
- Las medidas en la imagen en general se ven de igual modo influenciadas por el error de estimación de movimiento y presentan un error que también es creciente a medida que aumenta el índice temporal. (véase los elementos de la diagonal principal de la matriz Σ_L en (5.45)).

El efecto de considerar diferentes alternativas para la función de coste se muestra a continuación:

- Matriz identidad (M.I): el uso de la matriz identidad en la función de coste, equivale a medir la distancia una a una entre el conjunto de medidas y aquellas estimadas dada una solución para X_0^a :

$$\epsilon = (y_1^1 - \hat{y}_1^1)^2 + \dots + (y_3^3 - \hat{y}_3^3)^2 \quad (5.47)$$

El algoritmo de optimización minimiza por igual todos y cada uno de los residuos $(y_k^j - \hat{y}_k^j)^2$, sin importar las consecuencias que ello conllevan. Puesto que el error esperado en las medidas correspondientes al final de la secuencia es mayor debido a la odometría, la solución X_0^a se alterará de tal manera que se consiga disminuir dicho

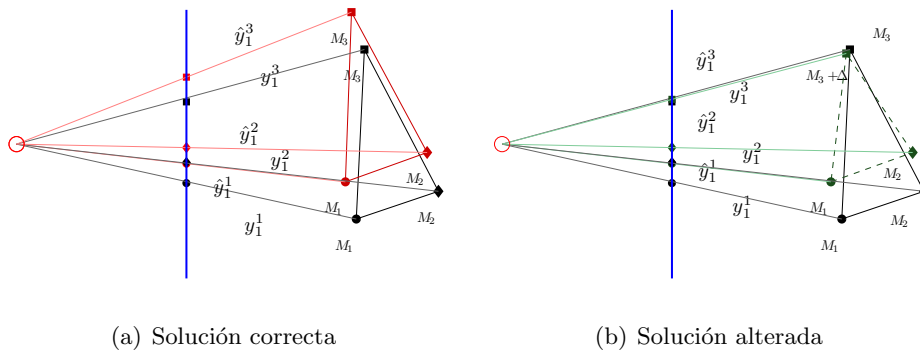
error, lo que explica con cierta claridad el fenómeno expuesto en el cuadro 5.1. En ese caso, la modificación de la geometría del robot mediante Δ_{X^a} mejora la distancia entre las medidas y la estimación de movimiento, a pesar de modificar la solución exacta. Si resultara que los sensores de odometría fueran mejores que lo estimado, la solución diagonal sería una medida justa ya que no existiría ninguna razón particular para creer que las últimas medidas contienen más error que aquellas más cercanas al inicio de la trayectoria de inicialización.

- Matriz Σ_L diagonal (M.D): en este caso, la función de coste consta de una suma ponderada de residuos independientes:

$$\epsilon = w_1(y_1^1 - \hat{y}_1^1)^2 + \dots + w_9(y_3^3 - \hat{y}_3^3)^2, \quad (5.48)$$

donde los pesos w_i corresponden a la inversa de los elementos de la diagonal principal de la matriz Σ_L . En este caso, la ponderación mide la influencia de cada uno de los residuos en la solución final. Aquellos con menor peso tendrán menor influencia, o lo que es lo mismo, aquellos para los que la varianza esperada es mayor. Este hecho es una de las principales causas por las cuales mediante el método diagonal, la solución errónea $X_0^a + \Delta_{X^a}$, en el ejemplo propuesto con anterioridad, obtenía mayor error que el valor correcto de X_0^a .

- Matriz Σ_L diagonal por bloques (M.B): en este caso, además de conservar los elementos de la diagonal principal, la matriz Σ_L aporta información acerca de la correlación mutua que tienen las medidas correspondientes a un mismo instante de tiempo. La correlación mutua se describe como bloques de 3x3 (3 puntos del triángulo medidos en el plano imagen), alrededor de la diagonal principal. Para observar el efecto adicional, que tiene el hecho de considerar la correlación cruzada, se diseña el experimento de la figura 5.5.



	Solución correcta X_0^a	Solución alterada $X_k^a + \Delta_{X^a}$
Aproximación por bloques (M.B.)	$\epsilon = 3,0581$	$\epsilon = 4,2080$
Aproximación diagonal (M. D.)	$\epsilon = 6,4336$	$\epsilon = 3,4581$
Matriz identidad (M.I.)	$\epsilon = 42,1770$	$\epsilon = 21,1506$

Figura 5.5: Diferencias entre compatibilidad individual y conjunta

En la figura 5.5.a se muestran las medidas estimadas (color rojo) comparadas con las reales (color negro) en el caso de utilizar la solución correcta X_0^a . En la figura 5.5.b, se muestran de nuevo la estimación de las medidas (color verde) en comparación con las obtenidas realmente (color negro). Se pretende mostrar el efecto que tiene sobre el coste del error, el hecho de considerar correlación entre las distintas medidas, y por ello el ejemplo se centra en un instante de tiempo $k = 1$.

En la tabla adjunta a la figura 5.5 se observa cómo el método diagonal y el método de matriz identidad fallan en este caso, dando lugar a un coste de error mayor en el caso de la solución incorrecta. Se observa también cómo la deformación introducida en el punto M^3 en la figura 5.5.b, estima una posición $\hat{y}_1^3$ en la imagen que es prácticamente solidaria con la medida real y_1^3 , mientras que los residuos $(y_1^1 - \hat{y}_1^1)^2$ y $(y_1^2 - \hat{y}_1^2)^2$ poseen el mismo error individual que la solución correcta del caso (a). La ponderación diagonal ejercida por el método M.D. en este caso no tiene efecto significativo sobre el coste del error, ya que por separado los residuos individuales son inferiores a los de la solución correcta. Sin embargo se observa como en el caso del método por bloques (M.B.), el coste asociado a la solución real sí es menor que el coste asociado a la solución errónea. Las correlaciones mutuas existentes en el bloque 3x3 de la matriz Σ_L son suficientes para determinar que la configuración (b), a pesar de presentar errores inferiores o iguales por separado en cada medida, es en realidad más errónea que la solución (a), ya que supone un triángulo deformado.

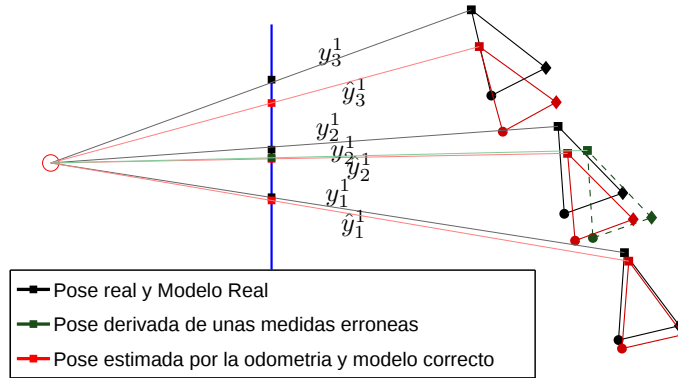
La principal justificación, por la cual el error utilizando el método M.B. es capaz de acertar en atribuir a la situación de la figura 5.5.b un coste menor que la situación de la figura 5.5.a, se justifica mediante la teoría de la compatibilidad estadística de procesos. Se considera al conjunto de medidas estimadas un proceso aleatorio gaussiano de media $(\hat{y}_1^1, \hat{y}_1^2, \hat{y}_1^3)^T$ y matriz de covarianza Σ_L . El coste atribuido a las medidas reales $(y_1^1, y_1^2, y_1^3)^T$ corresponde a la compatibilidad estadísticas de éstas con respecto al proceso. Si la matriz Σ_L es diagonal, ya sea identidad o no, cada medida es independiente estadísticamente de las demás, y por tanto la compatibilidad estadística es una suma individual de errores ponderados por los elementos de la diagonal principal de Σ_L . Sin embargo, si la matriz Σ_L no es diagonal, las correlaciones entre las distintas medidas establecen restricciones globales a la hora de comparar las medidas reales o realización del proceso con respecto a la media del proceso. Dichas restricciones globales, de forma intuitiva, establecerían que para que el conjunto de medidas sea compatible estadísticamente con el proceso, éstas deben presentar la misma relación espacial entre ellas. De este modo en el caso de la figura 5.5.a, como las medidas reales pertenecen al mismo triángulo que las estimaciones, son a pesar del error de la odometría una realización probable en conjunto del proceso. Sin embargo, en el caso de la figura 5.5.b, puesto que el modelo del robot tiene otra geometría que las medidas, aún y a pesar de que una de las medidas acierte, la distancia entre las otras dos no está correlada con el escaso error que presenta la tercera. En otras palabras, en conjunto las medidas no son una realización probable del proceso estimado, lo que da lugar a un coste superior.

Las limitaciones de la compatibilidad estadística, radican en el hecho de que los procesos son supuestos gaussianos en la vecindad de las condiciones actuales y por tanto la veracidad del test de compatibilidad sólo es aplicable en ese entorno. Sin

embargo, en esta tesis se muestra su utilidad en el proceso de reconstrucción para generar la función de coste de máxima verosimilitud.

- Matriz Σ_L completa (M.C.): la matriz de correlación completa incluye la relación estadística que existe no solo entre las distintas medidas de un mismo instante de tiempo, sino también la relación temporal entre ellas. La trayectoria que predice la odometría es una restricción para el proceso gaussiano equivalente de media $\hat{Y}_L$ y matriz de covarianza Σ_L . El tipo de situaciones en las que el uso de la matriz completa es necesaria se observa en la figura 5.6.

En ella se muestra una secuencia de tres instantes de tiempo, comparándose las medidas estimadas (en color rojo) con dos grupos de medidas diferenciados. El primer grupo (en color negro) corresponde a las medidas reales obtenidas con la solución correcta X_0^a . El segundo grupo coincide con el primero salvo en el instante $k = 2$, que se muestra en línea verde punteada. Se puede observar como en ese preciso instante las medidas correspondientes a la línea punteada son más cercanas al valor estimado de la posición del robot. Debido a este hecho, el método diagonal por bloques presenta inevitablemente un error menor en $k = 2$ y el mismo que con las medidas reales en el resto de instantes de tiempo. Como se puede observar en la tabla adjunta a la figura 5.6, el uso de la matriz completa es capaz de discernir que las medidas alteradas del instante $k = 2$ son menos probables para el proceso completo Y_L , ya que no son coherentes con el resto de la secuencia medida.



(a) Proyección en la línea imagen

	Medidas correctas	Medidas alteradas en k=2
Matriz completa (M.C.)	$\epsilon = 5,8501$	$\epsilon = 6,0396$
Aproximación por bloques (M.B.)	$\epsilon = 7,7262$	$\epsilon = 3,2468$
Aproximación diagonal (M. D.)	$\epsilon = 13,1293$	$\epsilon = 8,4991$
Matriz identidad (M.I.)	$\epsilon = 172,9449$	$\epsilon = 136,2753$

(b) Comparación de los errores de las funciones de coste utilizadas

Figura 5.6: Experimento que muestra el efecto del uso de la matriz completa en la función de coste.

Cada una de las propiedades mostradas para los ejemplos representativos utilizados, se puede fácilmente extrapolar al caso de tener una cámara bidimensional en la que se proyecta el mundo tridimensional, como es el caso del problema real de la tesis. Como se ha comentado con anterioridad, las diferentes alternativas propuestas para la inicialización serán defendidas con experimentos representativos utilizando configuraciones reales, dada la complejidad de asegurar cuestiones como la convergencia del algoritmo a un mínimo global.

5.2.5. Tiempo de cómputo

El mecanismo de inicialización, expuesto en el algoritmo 5, requiere en la práctica un número de iteraciones determinado para obtener un mínimo de la función de coste. En esta tesis se plantea en esencia aproximar la función de coste como cuadrática y utilizar el método de Newton para descender hasta el mínimo. Los métodos de tipo Newton (o métodos de segundo orden), poseen propiedades asintóticas de segundo orden si la aproximación cuadrática de la función de coste es suficientemente precisa. Cerca del mínimo de la función de coste, la precisión de la solución avanza doblando el número de dígitos con cada iteración.

Sin embargo, la bondad de los métodos cuadráticos también conlleva ciertas desventajas, que son detalladas a continuación:

- En muchas ocasiones la matriz Hessiana de la función de coste carece de inversa, bien porque el algoritmo se encuentra lejos del mínimo, o bien debido a la propia estructura del problema. En otros casos, el paso calculado mediante el método de Newton da lugar a mayor error que el calculado en la anterior iteración, lo cual quiere decir que la aproximación cuadrática de la función de coste no es realmente precisa o aplicable en ese punto.

En esos casos el incremento aditivo para los parámetros propuesto por los métodos de Newton, no es preciso y por ello se recurren a métodos de control, como puede ser el que se propone en esta tesis de Levenberg-Marquardt. Mediante este método, el cual es mostrado explícitamente en el algoritmo 5, la matriz Hessiana es modificada mediante la adición de una matriz diagonal ponderada por un parámetro unidimensional λ . Mediante dicho parámetro, el método de Newton se puede transformar en uno de primer orden de descenso por el gradiente a uno de Newton. El criterio para elegir entre uno y otro será la singularidad de la matriz Hessiana o el incremento del error de coste entre una iteración y la siguiente.

- El segundo problema de los métodos de Newton radica en la complejidad numérica que existe en cada iteración del algoritmo.

El hecho de tener que calcular la inversa de la matriz Hessiana requiere en general una complejidad $\mathcal{O}(N_{\Phi}^3)$ en cada iteración, siendo N_{Φ} el tamaño de los parámetros a estimar. Aprovechando una posible estructura de tipo “disperso” de la matriz existen métodos más rápidos que disminuyen considerablemente la complejidad del método (ver [Triggs et al., 2000]). En añadido a la inversa de la matriz Hessiana, el cálculo

analítico de las derivadas de la función de coste pueden añadir un considerable gasto computacional en función de la no linealidad y estructura de dichas funciones.

El uso de las distintas alternativas planteadas para la función de coste tiene un impacto directo en la complejidad del algoritmo por dos razones fundamentales:

- En primer lugar, para el cálculo de la matriz Hessiana, es necesario calcular no solo la matriz Σ_L sino además su inversa. En general el tamaño de las medidas, N_Y es mucho mayor que el de los parámetros N_Φ , ya que suelen necesitarse muchas mas medidas que parámetros a estimar, y por tanto dicha inversa puede representar un problema de complejidad dominante. En la tabla 5.2 se muestra la complejidad algorítmica derivada del cálculo e inversa de la matriz Σ_L en las distintas alternativas planteadas.

Matriz Completa (M.C.)	$\mathcal{O}(N_Y^3)$
Aproximación por bloques (M.B.)	$\mathcal{O}(N_Y^3/K^2)$
Aproximación diagonal (M.D.)	$\mathcal{O}(N_Y)$

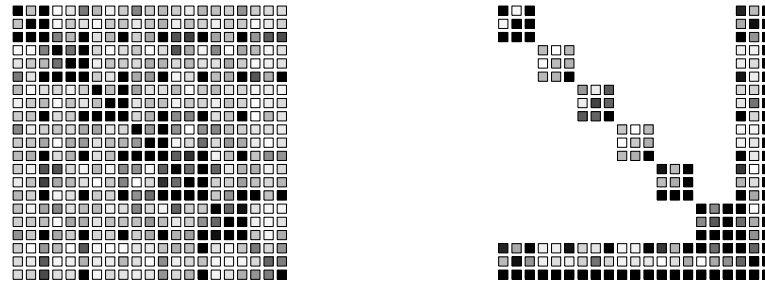
Tabla 5.2: Complejidad computacional de la inversa de la matriz Hessiana en las distintas alternativas de la función de coste Σ_L

- En añadido a la propia complejidad computacional de obtener la inversa de Σ_L en sus distintas variantes, la estructura de la matriz Σ_L afecta directamente a la disparidad de la matriz Hessiana, y por tanto a la posibilidad de acelerar finalmente el cálculo de su inversa.

En la figura 5.7, se observan los distintos tipos de estructuras que dan lugar a la matriz Hessiana. En los casos en los que la matriz Hessiana es completa corresponden con las alternativas de matriz completa y aproximada por bloques. En dichos casos la complejidad para el cálculo e inversa de Σ_L es claramente dominante a obtener la inversa de H . Sin embargo en el caso de tener una matriz Σ_L diagonal la matriz H presenta una estructura de tipo “flecha”, por lo que es posible encontrar una solución para el incremento de los parámetros con una complejidad menor. En la literatura de “Bundle Adjustment” existe una amplia documentación de los métodos para aprovechar las estructuras especiales de matrices de valores dispersos o “sparse”. En concreto dada una matriz con la estructura de tipo “flecha” mostrada en la figura 5.7.b, el método de Schur [Hartley and Zisserman, 2003] permite obtener el incremento de los parámetros en $\mathcal{O}(N_\Phi^2)$. En este caso la complejidad de invertir Σ_L es muy baja ($\mathcal{O}(N_Y)$), y en la mayoría de los casos inferior al cálculo de la inversa de la matriz Hessiana.

5.2.6. Tratamiento de los “outliers”

Como ya se ha comentado en capítulos anteriores, el hecho de usar la apariencia local de cada punto detectado en la imagen como base para el proceso de seguimiento de marcas naturales, produce inevitables errores de correspondencia y por lo tanto elementos del vector Y_L que no corresponden con su respectivo punto de la estructura del robot. La



(a) Hessiano resultante de M.C. y M.B. (b) Hessiano resultante de M.D. y M.I.

Figura 5.7: Estructura del Hessiano de las distintas alternativas de función de coste

identificación de las falsas correspondencias o “outliers” en la literatura anglosajona, se plantea como imprescindible para obtener un valor de inicialización no influenciado por medidas erróneas.

El modelado estadístico realizado en las distintas alternativas para la función de coste establece que, dada una solución para X_0^a , el proceso de medida $\mathbf{Y}_L$ se ajusta a una distribución gaussiana, la cual será mas o menos fiel a la realidad según el caso. El criterio de Máxima Verosimilitud utilizado permite obtener una solución buscando el valor de X_0^a que minimice una discrepancia estadística entre la media del proceso $\hat{Y}_L$ y las medidas reales Y_L consideradas una posible realización del mismo.

El problema ocurre por tanto, cuando existen “outliers” en el vector Y_L que no pertenecen al proceso gaussiano estimado y se ajustan a otra distribución de probabilidad. Ése es el caso si accidentalmente el proceso de seguimiento confunde la identidad de puntos de la estructura del robot. En ese caso minimizar la distancia de Mahalanobis tiene como resultado una solución para X_0^a desviada o sesgada. El sesgo introducido puede ser más que suficiente para invalidar el proceso de inicialización.

Una solución muy extendida consiste en modificar la función de coste de modo que ésta reduzca la influencia de aquellos puntos que no son probables como realización de Y_L y por lo tanto son considerados “outliers”. En general la versión robusta de una función de coste consiste en determinar el mínimo grupo de residuos o discrepancias que son linealmente separables y aplicarles una ponderación que permita separar aquellos considerados “outliers”. Dicha ponderación es una función $\rho(\epsilon^2)$, siendo ϵ^2 un residuo concreto. La función ρ o estimador robusto tiene la facultad de anular los residuos que no se ajustan correctamente al problema y así su influencia en la solución.

Existen diversas propuestas de estimadores robustos, muchos de ellos ajustados a una posible distribución estadística de los “outliers” y otros como resultado de un análisis heurístico. En [Triggs et al., 2000] se muestran muchos de ellos y sus propiedades, así como una reducida bibliografía donde encontrar más información acerca de los mismos. En esta tesis se empleará un modelado de los “outliers” como una distribución de Cauchy, la cual modela de manera sencilla y analítica errores “impulsivos” dentro del vector Y_L :

$$\rho(\epsilon^2) = b^2 \log(1 + \epsilon^2/b^2) \quad (5.49)$$

La función ρ es no lineal y el efecto teórico de suprimir “outliers” se aplica a su argumento en totalidad. Esto quiere decir que toda la información involucrada en calcular cada residuo será globalmente penalizada como errónea o por el contrario influenciará en la solución final.

Matriz Completa (M.C.)	$\rho((Y_L - \hat{Y}_L)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L))$
Aproximación por bloques (M.B.)	$\sum_{k=1}^K \rho((Y_k - \hat{Y}_k)^T \Sigma_{Y_k}^{-1} (Y_k - \hat{Y}_k))$
Aproximación diagonal (M.D.)	$\sum_{k=1}^K \sum_{i=1}^N \rho((y_k^i - \hat{y}_k^i)^T \Sigma_{y_k^i}^{-1} (y_k^i - \hat{y}_k^i))$
Matriz identidad (M.I.)	$\sum_{k=1}^K \sum_{i=1}^N \rho((y_k^i - \hat{y}_k^i)^T (y_k^i - \hat{y}_k^i))$

Tabla 5.3: Versiones robustas de las distintas alternativas para la función de coste.

Como puede verse en la tabla 5.3, el bloque mínimo establecido para la aproximación diagonal de Σ_L como para la matriz identidad corresponde al error en las coordenadas horizontal y vertical de una medida concreta y_k^i . Éste suele ser el caso deseable ya que de este modo cada medida por separado es analizada en función del residuo individual que genera y puede ser eliminada sin afectar a medidas correctas correspondientes al mismo instante de tiempo o a la misma marca.

Sin embargo, la situación cambia cuando no existen grupos pequeños separables de residuos como en el caso de la aproximaciones por bloques de Σ_L o su versión completa.

En el caso de una aproximación de Σ_L por bloques, el mínimo residuo separable linealmente es cada bloque correspondiente a todas las medidas en un mismo instante de tiempo o Y_k . El uso de una versión robusta en este caso no es aconsejable puesto que podrían descartarse prácticamente todas las medidas en el caso de que una sola de ellas fuera errónea en la secuencia. En cuanto a la versión completa de Σ_L , no existe un grupo separable y por tanto el residuo es el bloque completo de información interrelacionada. La función robusta no puede ser aplicada del mismo modo en este caso ya que sólo puede ponderar al completo la función de coste.

Para las alternativas de matriz Σ_L en las que la separabilidad lineal de los residuos no es posible, el problema planteado se puede transformar de manera que la función robusta no afecte directamente a los residuos cuadráticos pero sí a cada vector resta $Y_L - \hat{Y}_L$.

En ese caso la nueva función de coste robusta será:

$$(Y_L - \hat{Y}_L)^T W^T \Sigma_L^{-1} W (Y_L - \hat{Y}_L), \quad (5.50)$$

siendo W la matriz diagonal de tamaño $N_Y \times N_Y$ dada por (5.51):

$$W = \begin{pmatrix} w_1^1 I_{2 \times 2} & \cdots & \bar{0} \\ \vdots & \ddots & \vdots \\ \bar{0} & \cdots & w_K^N I_{2 \times 2} \end{pmatrix}^T, \quad (5.51)$$

donde cada elemento de ponderación w_k^i , el cual afecta a su correspondiente residuo $\epsilon_k^i = y_k^i - \hat{y}_k^i$, es:

$$w_k^i = \frac{\sqrt{\rho((\epsilon_k^i)^2)}}{\sqrt{2}|\epsilon_k^i|}, \quad (5.52)$$

y ρ es la función robusta comentada en (5.42).

La matriz W está diseñada para que en ausencia de Σ_L la función de coste resultante sea equivalente a la correspondiente versión robusta, es decir:

$$(Y_L - \hat{Y}_L)^T W^T W (Y_L - \hat{Y}_L) = \sum_{k=1}^K \sum_{i=1}^N \rho((y_k^i - \hat{y}_k^i)^T (y_k^i - \hat{y}_k^i)) \quad (5.53)$$

Mediante esta propuesta es posible implementar una función robusta para cualquier versión de Σ_L , unificando, por tanto, la notación y las distintas funciones de coste de la tabla 5.3, en la expresión compacta (5.50).

Ajuste de la función robusta

La elección del parámetro “ b ”, mostrado en la ecuación (5.49), se realiza asumiendo que cada medida simple y_k^i , considerada como “inlier”, se describe como un proceso gaussiano con covarianza Σ_v . Por tanto cada residuo simple $(\epsilon_k^i)^2$, es una variable aleatoria χ^2 de dos grados de libertad. El valor del residuo $|\epsilon_k^i|$ que representa el 95 % de la distribución acumulativa de χ^2 es 2.44, de tal modo que residuos de mayor valor pueden empezar a considerarse “outliers” ya que su probabilidad de pertenecer a y_k^i es inferior a 0,05. En la figura 5.8, se muestra la evaluación de la función robusta de Cauchy para varios valores de b en comparación con la versión cuadrática no robusta. Un valor de $b \in [2, 3]$, se comporta como una función cuadrática para valores de $|\epsilon_k^i| < 2,44$, atenuando a través de la pendiente de la curva, casi plana, la influencia de los “outliers” debidos a errores de correspondencia.

5.2.7. Inicialización de la solución previa a la optimización

El método de inicialización propuesto en este capítulo, el cual obtiene la estimación de X_0^a mediante un algoritmo de optimización, requiere de un valor de inicio en la primera iteración de la optimización. Encontrar un valor cercano a la solución es relevante para que el método iterativo alcance el mínimo global de la función y no se desvíe a puntos de inflexión locales.

En general se busca un método no iterativo, a ser posible lineal, que permita de una manera directa alcanzar una primera estimación del vector X_0^a o una parte significativa del mismo.

Este apartado propone un método lineal que obtiene el valor de X_0^a , en el caso de tener una cámara proyectiva sin distorsión y un robot que se mueve en un plano.

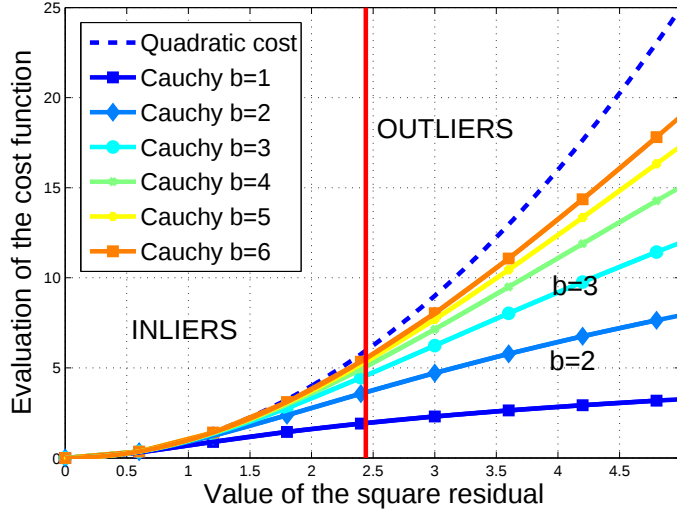


Figura 5.8: Valor de la distribución de Cauchy en función del parámetro “b”

En este apartado se concretan las transformaciones en el plano imagen y el movimiento del robot, de tal manera que se pueda presentar en detalle cada una de las propuestas.

Sea un punto M^i perteneciente a la estructura del robot con una pose X_k , entonces su proyección en el plano imagen y_k^i será:

$$\begin{pmatrix} y_k^i \\ 1 \end{pmatrix} = \lambda_k^i K_c (R_c (R_k M^i + T_k) + T_c), \quad (5.54)$$

donde λ_k^i representa el factor de escala que permite transformar las coordenadas homogéneas en el valor de y_k^i en el plano imagen, R_c y T_c y K_c representan los parámetros extrínsecos y intrínsecos de la cámara respectivamente. La matriz de rotación R_k y el vector de traslación T_k representan la pose del robot en el instante k en forma de transformación afín. Su relación con las variables del vector de estado $X_k = (x_k^1, x_k^2, \alpha_k)$, es la siguiente:

$$R_k = \begin{pmatrix} \cos(\alpha_k) & -\sin(\alpha_k) & 0 \\ \sin(\alpha_k) & \cos(\alpha_k) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad T_k = \begin{pmatrix} x_k^1 \\ x_k^2 \\ 0 \end{pmatrix} \quad (5.55)$$

De esta manera, la función $M_k^i = t(M^i, X_k)$, será:

$$t(M^i, X_k) = R_k M^i + T_k \quad (5.56)$$

Haciendo uso de los modelos de movimiento, la matriz R_k y el vector T_k se relacionan con los valores iniciales de la pose del siguiente modo:

$$R_k = R_k^\Delta R_0 \quad T_k = T_0 + R_0 T_k^\Delta, \quad (5.57)$$

donde R_k^Δ y T_k^Δ corresponden a la estimación que se obtiene de la odometría, suponiendo que el valor inicial de la pose es el vector nulo.

La expresión (5.54), en función de R_0 y T_0 será:

$$\begin{pmatrix} y_k^i \\ 1 \end{pmatrix} = \lambda_k^i K(R_c(R_k^\Delta R_0 M^i + T_0 + R_0 T_k^\Delta) + T_c). \quad (5.58)$$

El valor de y_k^i obtenido es una función no lineal de los parámetros X_0 y M^i que se quieren calcular y que componen el vector inicial X_0^a . La no linealidad en este caso proviene de dos fuentes fundamentales:

- En primer lugar el valor de R_0 es una función no lineal del ángulo α_0 . Además, dicha matriz aparece multiplicando a los valores de geometría del robot M^i , los cuales también son parte del vector X_0^a .
- La segunda fuente de no linealidad proviene de la función de proyección en el plano imagen, la cual requiere el cálculo del parámetro λ_k^i , igual a la inversa de la tercera coordenada obtenida con la transformación homogénea.

Se propone una reparametrización de X_0^a y una reestructuración de la función de proyección que permita evitar la inversa implícita en el parámetro λ . En primer lugar, puesto que tanto la matriz R_0 como los parámetros M^i no varían con el tiempo, se puede incluir en el vector de estados los puntos rotados $M_0^i = R_0 M^i$, eliminando así la no linealidad del producto entre parámetros. Una vez calculados, el conjunto original se obtendría fácilmente invirtiendo la rotación.

Para evitar las funciones trigonométricas, el parámetro α_0 será sustituido por un vector bidimensional unitario (a, b) , donde $a = \cos(\alpha_0)$ y $b = \sin(\alpha_0)$. En todo caso se deberá tener en cuenta que a y b tienen la condición de contorno, $a^2 + b^2 = 1$.

Utilizando dicha parametrización, la expresión (5.54) se transforma en la siguiente:

$$\begin{pmatrix} y_k^i \\ 1 \end{pmatrix} = \lambda K_c(R_c(R_k^\Delta M_0^i + T_0 + L_{T_k^\Delta} \begin{pmatrix} a \\ b \\ 0 \end{pmatrix}) + T_c), \quad (5.59)$$

siendo $L_{T_k^\Delta}$:

$$L_{T_k^\Delta} = \begin{pmatrix} x_k^{1,\Delta} & -x_k^{2,\Delta} & 0 \\ -x_k^{2,\Delta} & x_k^{1,\Delta} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (5.60)$$

donde $x_k^{1,\Delta}$ y $x_k^{2,\Delta}$ son las dos primeras coordenadas del vector T_k^Δ .

El nuevo vector de estados con la reparametrización será:

$$\Phi = (x_0^1 \quad x_0^2 \quad a \quad b \quad (M^1)^T, \dots, (M^N)^T)^T \quad (5.61)$$

Desglosando en forma vectorial la transformación proyectiva se obtiene:

$$\begin{pmatrix} u_k^i \\ v_k^i \end{pmatrix} = y_k^i \quad \begin{pmatrix} U_k^i \\ V_k^i \\ S_k^i \end{pmatrix} = K(R_c(R_k^\Delta M_0^i + T_0 + L_{T_k^\Delta}) + T_c), \quad (5.62)$$

donde gracias a la nueva parametrización las componentes U_k^i , V_k^i y S_k^i equivalen a una transformación lineal de los parámetros Φ :

$$U_k^i = L_{U_k}^i \Phi + b_{U_k}^i \quad V_k^i = L_{V_k}^i \Phi + b_{V_k}^i \quad z_k^i = L_{S_k}^i \Phi + b_{S_k}^i \quad (5.63)$$

Para evitar el parámetro λ_k^i en la transformación en el plano imagen, y así poder obtener relaciones lineales, se utiliza la siguiente propiedad del producto vectorial:

$$\begin{pmatrix} u_k^i \\ v_k^i \\ 0 \end{pmatrix} = \lambda_k^i \begin{pmatrix} U_k^i \\ V_k^i \\ S_k^i \end{pmatrix} \rightarrow \begin{pmatrix} u_k^i \\ v_k^i \\ 0 \end{pmatrix} \times \begin{pmatrix} U_k^i \\ V_k^i \\ S_k^i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad (5.64)$$

de donde se obtienen tres relaciones lineales con los parámetros incluidos en el vector Φ , de las cuales tan solo dos son linealmente independientes:

$$\begin{cases} v_k^i S_k^i - V_k^i = 0 \\ -u_k^i S_k^i + U_k^i = 0 \end{cases} \rightarrow \begin{cases} (v_k^i L_{S_k}^i - L_{V_k}^i) \Phi = v_k^i b_{S_k}^i - b_{V_k}^i \\ (-u_k^i L_{S_k}^i + L_{U_k}^i) \Phi = -u_k^i b_{S_k}^i + b_{U_k}^i \end{cases} \quad (5.65)$$

De este modo, cada medida en el plano imagen aporta dos soluciones linealmente independientes y, dado el conjunto de medidas Y_L correspondientes a la trayectoria de inicialización, se obtiene un sistema lineal, en general sobredeterminado, para la obtención de Φ :

$$A\Phi = B \quad A = \begin{pmatrix} (v_1^1 L_{S_1}^1 - L_{V_1}^1) \\ (-u_1^1 L_{S_1}^1 + L_{U_1}^1) \\ \vdots \\ (v_K^N L_{S_K}^N - L_{V_K}^N) \\ (-u_K^N L_{S_K}^N + L_{U_K}^N) \end{pmatrix} \quad B = \begin{pmatrix} v_1^1 b_{S_1}^1 - b_{V_1}^1 \\ -u_1^1 b_{S_1}^1 + b_{U_1}^1 \\ \vdots \\ v_K^N b_{S_K}^N - b_{V_K}^N \\ -u_K^N b_{S_K}^N + b_{U_K}^N \end{pmatrix} \quad (5.66)$$

Se puede demostrar fácilmente que el rango de A es inferior al tamaño de Φ , consecuentemente tiene al menos un autovalor nulo, para cualquier trayectoria llevada a cabo por el robot y con cualquier numero de puntos y configuración de los mismos.

Para demostrar que existe al menos un autovalor de la matriz $A^T A$ que es nulo, se propone mostrar una familia paramétrica de soluciones para el vector Φ que dan lugar a proyecciones en la imagen Y_L idénticas.

Dado el conjunto de medidas $\{y_k^i \mid i = 1, \dots, N \ k = 1, \dots, K\}$ es generado por los valores de $\Phi = (x_0^1 \ x_0^2 \ a \ b \ M_0^1 \ \dots \ M_0^N)$ siguiendo la expresión (5.59):

$$\begin{aligned}
\begin{pmatrix} y_1^1 \\ 1 \end{pmatrix} &= \lambda K_c (R_c (R_k^\Delta M_0^1 + T_0 + L_{T_1^\Delta} \begin{pmatrix} a \\ b \\ 0 \end{pmatrix}) + T_c) \\
&\vdots \quad \vdots \quad \vdots \\
\begin{pmatrix} y_K^N \\ 1 \end{pmatrix} &= \lambda K_c (R_c (R_k^\Delta M_0^N + T_0 + L_{T_K^\Delta} \begin{pmatrix} a \\ b \\ 0 \end{pmatrix}) + T_c).
\end{aligned} \tag{5.67}$$

Se demuestra fácilmente que si el vector Φ_0 es solución, entonces $\Phi_0 + \psi\Delta$ también lo es en cualquier caso y para cualquier $\psi \in \mathbb{R}$, siendo Δ :

$$\Delta = \begin{pmatrix} T_{c,x^1} & T_{c,x^2} & a & b & \begin{pmatrix} 0 \\ 0 \\ T_{c,x^3} \end{pmatrix}^T & \cdots & \begin{pmatrix} 0 \\ 0 \\ T_{c,x^3} \end{pmatrix}^T \end{pmatrix} \tag{5.68}$$

De hecho, normalizando Δ a un vector unitario Δ_u , se obtiene el autovector de $A^T A$ asociado al autovalor nulo. Cabe destacar que el hecho de que exista una familia de soluciones para el vector Φ no proviene realmente de la falta de restricciones sobre Φ , sino que es producido por el aumento de los grados de libertad en la reparametrización. Al no imponerse restricciones de contorno para a y b , cabe la posibilidad de que el algoritmo entregue un valor que no cumpla que $a^2 + b^2 = 1$. De hecho, la familia de soluciones esta relacionadas por un factor de escala en a y b y como veremos a continuación, fijando dicho valor de escala la ambigüedad es resuelta.

Se propone a continuación un sencillo método sistemático para encontrar la familia de soluciones gobernada por un parámetro, que será calculado a “posteriori” utilizando la condición de contorno $a^2 + b^2 = 1$.

Sea la descomposición en valores singulares de $A^T A$, formada por la matriz ortogonal U y la matriz diagonal D :

$$A^T A = U D U^T \quad D = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_{N_\Phi} \end{pmatrix} \quad U = \begin{pmatrix} U_{\lambda_1} & \cdots & U_{\lambda_{N_\Phi}} \end{pmatrix} \tag{5.69}$$

Cada columna U_{λ_i} de la matriz U contiene el autovector asociado al valor singular λ_i , y por tanto U_{λ_1} corresponde con el vector unitario Δ_u que parametriza la familia de infinitas soluciones del sistema. La matriz U es ortogonal y por tanto su inversa existe siempre. Despejando U en la expresión (5.66), se obtiene el siguiente sistema:

$$D \begin{pmatrix} U_{\lambda_1}^T \\ \vdots \\ U_{\lambda_{N_\Phi}}^T \end{pmatrix} \Phi = U^T A^T B. \tag{5.70}$$

A su vez, la matriz A , la cual es de dimensiones $N_y \times N_\Phi$, tiene la siguiente descomposición en valores singulares:

$$A = V D_A U^T \quad V^T V = I_{N_\Phi \times N_\Phi} \quad D_A^T D_A = D, \quad (5.71)$$

donde la matriz U es la misma obtenida en la diagonalización de $A^T A$. Haciendo uso de éste último resultado, el término independiente $U^T A^T B$ será:

$$U^T A^T B = D_A V^T B = \begin{pmatrix} 0 \\ B^p \end{pmatrix} \quad (5.72)$$

Puesto que la matriz U^T es ortogonal y de las mismas dimensiones que el vector Φ , sus vectores en realidad forman una base del espacio de dimensión N_Φ y por tanto el vector Φ se puede expresar como una combinación lineal de los vectores de dicha base:

$$\Phi = \kappa_{\lambda_1} U_{\lambda_1} + \cdots + \kappa_{\lambda_{N_\Phi}} U_{\lambda_{N_\Phi}} \quad (5.73)$$

Introduciendo dicha representación en la expresión (5.70), se obtiene una solución para los coeficientes $\kappa_{\lambda_2}, \dots, \kappa_{\lambda_{N_\Phi}}$:

$$\begin{pmatrix} \kappa_{\lambda_2} \\ \vdots \\ \kappa_{\lambda_{N_\Phi}} \end{pmatrix} = \begin{pmatrix} 1/\lambda_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1/\lambda_{N_\Phi} \end{pmatrix} B^p \quad (5.74)$$

La familia paramétrica de soluciones para Φ se obtiene, utilizando como solución cualquiera Φ_0 la otorgada por los coeficientes $\kappa_{\lambda_2}, \dots, \kappa_{\lambda_{N_\Phi}}$, es decir:

$$\Phi_0 = \kappa_{\lambda_2} U_{\lambda_2} + \cdots + \kappa_{\lambda_{N_\Phi}} U_{\lambda_{N_\Phi}} \quad (5.75)$$

$$\Phi = \Phi_0 + \kappa_{\lambda_1} U_{\lambda_1} \quad (5.76)$$

Para encontrar el valor de κ_{λ_1} , correspondiente a la solución real, se impone la restricción $a^2 + b^2 = 1$.

Llamando a_u y b_u a las componentes correspondientes a a y b del vector U_{λ_1} , se obtiene una ecuación cuadrática para solucionar el valor de κ_{λ_1} :

$$\kappa_{\lambda_1}^2 + \left(\frac{2a_u a + 2b_u b}{a_u^2 + b_u^2} \right) \kappa_{\lambda_1} + \frac{a^2 + b^2 - 1}{a_u^2 + b_u^2} = 0 \quad (5.77)$$

De la expresión (5.77) se obtienen dos soluciones para el valor de κ_{λ_1} . Una de ellas da lugar a una solución ficticia que se proyecta en la cámara en direcciones del eje Z de la cámara negativas. Para encontrarla, tan solo hay que calcular la posición del centro del robot desde el origen de coordenadas de la cámara, es decir $R_c T_0 + T_c$, y observar para cual de las opciones de κ_{λ_1} se obtiene una tercera coordenada negativa.

Una vez obtenido el valor de κ_{λ_1} de la solución correcta, el valor de Φ único se obtiene particularizando la familia de soluciones $\Phi = \Phi_0 + \kappa_{\lambda_1} U_{\lambda_1}$.

La solución buscada del vector X_0^a requiere invertir la parametrización utilizada:

$$\alpha_0 = \tan^{-1}(a, b) \quad M^i = R_0^{-1} M_0^i \quad X_0^a = (x_0^1 \ x_0^2 \ \alpha_0 \ (M^1)^T \ \dots \ (M^N)^T)^T. \quad (5.78)$$

El método lineal propuesto es por tanto exacto y sólo requiere calcular una vez la descomposición en valores singulares de $A^T A$. Al ser exacto, incluso en los parámetros de rotación, entregará una estimación al algoritmo de optimización que en pocas iteraciones será ajustado con precisión.

5.2.8. Obtención de la distribución gaussiana de la solución

Una vez obtenida la solución X_0^a a la inicialización, utilizando el método propuesto en este capítulo, es necesario obtener a su vez la matriz de covarianza Σ_0^a que describe la dispersión del error obtenido en la optimización. A su vez, la obtención de Σ_0^a se hace necesaria de tal modo que el método de inicialización conecte con el método secuencial que se detalla en el siguiente capítulo, en el que todos los procesos son descritos con funciones de densidad de probabilidad gaussianas.

La metodología utilizada para caracterizar la varianza de X_0^a consiste en suponer que una vez finalizado el proceso de optimización descrito en el Algoritmo 5, la solución obtenida se encuentra contaminada con un ruido aleatorio Δ_{X^a} , correspondiente al incremento obtenido para X_0^a alrededor del mínimo calculado:

$$\Delta_{X^a} = -(H)^{-1} J \quad H = 2 \left(\frac{\partial \hat{Y}_L}{\partial X_0^a} \right)^T \Sigma_L^{-1} \left(\frac{\partial \hat{Y}_L}{\partial X_0^a} \right) \quad J = -2 \left(\frac{\partial \hat{Y}_L}{\partial X_0^a} \right)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L) \quad (5.79)$$

Si se ha obtenido el mínimo de la función de coste, se cumple que $E[\Delta_{X^a}] = 0$ y su varianza, la cual coincide con Σ_0^a , será:

$$\Sigma_0^a = E[\Delta_{X^a} \Delta_{X^a}^T] = (H)^{-1} J J^T H^{-T} \quad (5.80)$$

Definiendo como J_{X^a} a la expresión $\frac{\partial \hat{Y}_L}{\partial X_0^a}$:

$$\Sigma_0^a = (J_{X^a}^T \Sigma_L^{-1} J_{X^a})^{-1} J_{X^a}^T \Sigma_L^{-1} E[(Y - \hat{Y}_L)(Y - \hat{Y}_L)^T] \Sigma_L^{-1} J_{X^a} (J_{X^a}^T \Sigma_L^{-1} J_{X^a})^{-1} \quad (5.81)$$

Sustituyendo $E[(Y - \hat{Y}_L)(Y - \hat{Y}_L)^T] = \Sigma_L$, la expresión (5.81) se simplifica en la siguiente expresión compacta:

$$\Sigma_0^a = (J_{X^a}^T \Sigma_L^{-1} J_{X^a})^{-1} \quad (5.82)$$

5.2.9. Configuraciones degeneradas

El método de inicialización propuesto en este capítulo requiere que el robot realice una trayectoria de inicialización con el objetivo de utilizar el movimiento generado para reconstruir la geometría y pose del robot. Sin embargo, existen situaciones en las que el movimiento realizado por el robot da lugar a múltiples soluciones para el vector X_0^a . Existen tres tipos fundamentales de trayectorias en las que se demuestra con facilidad que dan lugar a múltiples soluciones:

- Ausencia de velocidad angular: dada una trayectoria rectilínea, no existen suficientes restricciones para encontrar un punto del espacio en el que fijar el origen de coordenadas del robot, dando lugar a una familia infinita de posibles reconstrucciones de la geometría del robot, de las cuales sólo una será correcta en el caso de que el robot realice una rotación.

En estas circunstancias, un punto cualquiera de la geometría M^i del robot se proyecta en la imagen en el instante k , dando lugar al punto en el plano imagen y_k^i :

$$y_k^i = K(R_c(R_0 M^i + T_0 + R_0 \begin{pmatrix} x^{1,\Delta} \\ 0 \\ 0 \end{pmatrix})) \quad T_0 = \begin{pmatrix} x_0^1 \\ x_0^2 \\ 0 \end{pmatrix}, \quad (5.83)$$

donde, al tratarse de una trayectoria rectilínea, la posición dada por la odometría, normalizada a la orientación inicial, tan solo posee componentes en una de sus coordenadas ($x^{1,\Delta}$). Se demuestra fácilmente cómo la siguiente familia biparamétrica de soluciones $X_0^a(\Delta_1, \Delta_2)$ da lugar a las mismas medidas en el plano imagen:

$$M_\Delta^i = M^i + R_0^{-1} \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ 0 \end{pmatrix} \quad T_{0\Delta} = T_0 - \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ 0 \end{pmatrix} \quad (5.84)$$

$$X_0^a(\Delta_1, \Delta_2) = (T_{0\Delta} \quad \alpha_0 \quad M_\Delta^1 \quad \cdots \quad M_\Delta^N) \quad (5.85)$$

- Ausencia de velocidad lineal: dado un movimiento compuesto únicamente por consignas de velocidad angular, el origen de rotación del robot permanece constante mientras cada punto de la estructura del robot describe una circunferencia alrededor del eje de rotación definido por T_0 .

En este caso un punto M^i del robot se proyecta en la imagen en el instante k , dando lugar al punto en el plano imagen y_k^i mediante la siguiente expresión:

$$y_k^i = K(R_c(R_k^\Delta R_0 M^i + T_0) + T_c), \quad (5.86)$$

Se demuestra que la siguiente familia uniparamétrica de soluciones $X_0^a(n)$ da lugar a medidas idénticas en el plano imagen:

$$M_n^i = nM^i + (n-1) \begin{pmatrix} 0 \\ 0 \\ \bar{T}_c(3) \end{pmatrix} \quad T_{0n} = nT_0 + (n-1) \begin{pmatrix} \bar{T}_c(1) \\ \bar{T}_c(2) \\ 0 \end{pmatrix}, \quad (5.87)$$

siendo $\bar{T}_c = R_c^T T_c$.

$$X_k^a(n) = (T_{0n} \quad \alpha_0 \quad M_n^1 \quad \cdots \quad M_n^N) \quad (5.88)$$

- Trayectoria circular: dada una trayectoria en la que la posición del robot describe una circunferencia de radio ρ , la proyección de un punto M^i de la geometría del robot se determina por:

$$y_k^i = K(R_c(R_k^\Delta R_0 M^i + T_0 + R_0 \begin{pmatrix} \rho \cos(\alpha_k^\Delta) + x_c \\ \rho \sin(\alpha_k^\Delta) + x_c \\ 0 \end{pmatrix}) + T_c), \quad (5.89)$$

Se demuestra que la siguiente familia uniparamétrica de soluciones $X_0^a(n)$ da lugar a medidas idénticas en el plano imagen:

$$M_n^i = nM^i + R_0^T \quad T_{0n} = nT_0 + (n-1)R_c^T T_c \quad (5.90)$$

$$X_k^a(n) = (T_{0n} \quad \alpha_0 \quad M_n^1 \quad \cdots \quad M_n^N) \quad (5.91)$$

Como se ha demostrado en este apartado, los movimientos puramente rectilíneos o aquellos puramente circulares corresponden con situaciones degeneradas, en las cuales se pueden encontrar múltiples soluciones al problema planteado. De este modo, se debe evitar este tipo de movimientos en la trayectoria de inicialización realizada por el robot. La mejor estrategia resulta en la práctica de una combinación de trayectorias curvas con tramos rectos, con el fin de no acercarse a una configuración degenerada.

5.2.10. Evaluación experimental del método propuesto

Con el objetivo de evaluar las distintas alternativas propuestas para el mecanismo de inicialización mediante una cámara, se plantea la realización de un conjunto de experimentos basados en datos artificiales, lo cual permite evaluar los errores cometidos ante la variación de diferentes parámetros de interés.

El entorno de prueba, utilizando una cámara, se basará en la distribución geométrica de una situación realista, sobre la que se realizarán pruebas reales en el capítulo 8. A continuación se resumen las características del entorno de prueba, el cual se describe con más detalle en el citado capítulo:

- Se dispone de una cámara “pin-hole” con parámetros intrínsecos

$$(f_u = 636,7888, f_v = 637,5610, u_0 = 313,3236, v_0 = 210,6894),$$

y la posición y orientación con respecto al origen del mundo O_W que se muestra en la figura 5.9. La matriz Σ_V , que modela el ruido de medida, se supone diagonal con un valor de $\sigma_v^2 = 10$ en su diagonal principal.

- El robot consta de un conjunto de $N = 10$ puntos distribuidos aleatoriamente dentro de un cilindro de radio $R = 0,5m$ y altura $h = 1m$, lo cual corresponde con la altura máxima que tendrá el tipo de robot utilizado.

- Se ha simulado un sistema de odometría en el robot en el que la incertidumbre en las lecturas de odometría se representa mediante la matriz diagonal Σ_W , definida por las varianzas del ruido de la lectura de velocidad lineal y angular $\sigma_{V_l}^2 = 10$ y $\sigma_{\Omega}^2 = 1$ respectivamente.
- La trayectoria de inicialización utilizada en los experimentos se visualiza en la figura 5.9, y consta de un tramo rectilíneo de $3m$ de longitud y media circunferencia de $1m$ de radio.
- Por simplicidad se supone que todos los puntos del robot son visibles en todas las imágenes y que no existen oclusiones en los mismos. Para comprobar la inmunidad ante la presencia de “outliers”, se utiliza una distribución realista que concuerde con la distribución real de los mismos. Los outliers se generarán a partir de una probabilidad p_{out} con respecto al total de puntos medidos y contenidos en el vector Y_L . Mediante dicha probabilidad, se eligen que puntos del vector Y_L corresponden con “outliers”, siendo sustituidos por tres fuentes de ruido diferentes: la primera fuente de ruido se compone de puntos de la estructura del robot que no forman parte del conjunto de $N = 10$ original y que consecuentemente se mueven junto con el robot y modelan los errores de “matching” con puntos del robot. El segundo tipo de fuente de error consiste en la utilización de puntos fijos en el entorno y que teóricamente corresponderían con falsas correspondencias con puntos del fondo de la escena. La tercera fuente de error se modelo como un ruido aleatorio gaussiano que se añade a la correspondencia en cuestión. Dicho ruido tiene una varianza en el plano imagen representada por la matriz $\Sigma_{V,o}$ cuya diagonal principal tiene un valor alto ≈ 1000 para representar ruido impulsivo de medida. Como se ha visto en el capítulo 4, esta última fuente de ruido es la menos probable debido al método de “tracking” supervisado que se utiliza. La cantidad de “outliers” que se introducen de cada parte se ha distribuido experimentalmente de tal forma que el 60 % de los “outliers” generados sean producidos por la primera fuente de error, el 30 % con puntos del fondo y el 10 % restante corresponde con la distribución impulsiva. Los porcentajes utilizados se corresponden con aquellos encontrados aproximadamente en situaciones reales.
- Los errores en la inicialización se dividen en tres magnitudes.
 1. Se define como error en la reconstrucción del modelo ϵ_M al error relativo resultante de la siguiente expresión:

$$\epsilon_M = \frac{\sqrt{\sum_{i=1}^N \|M^i - M_{gth}^i\|^2}}{\sqrt{\sum_{i=1}^N \|M_{gth}^i\|^2}}, \quad (5.92)$$

donde M_{gth}^i corresponde con los valores del modelo del robot utilizado como referencia para generar el ejemplo.

2. Se define como error en la pose estimada al error en posición (distancia euclídea) y el error en orientación (grados):

$$\epsilon_T = \|T_0 - T_{0,gth}\| \quad \epsilon_\alpha = \|\alpha_0 - \alpha_{0,gth}\|, \quad (5.93)$$

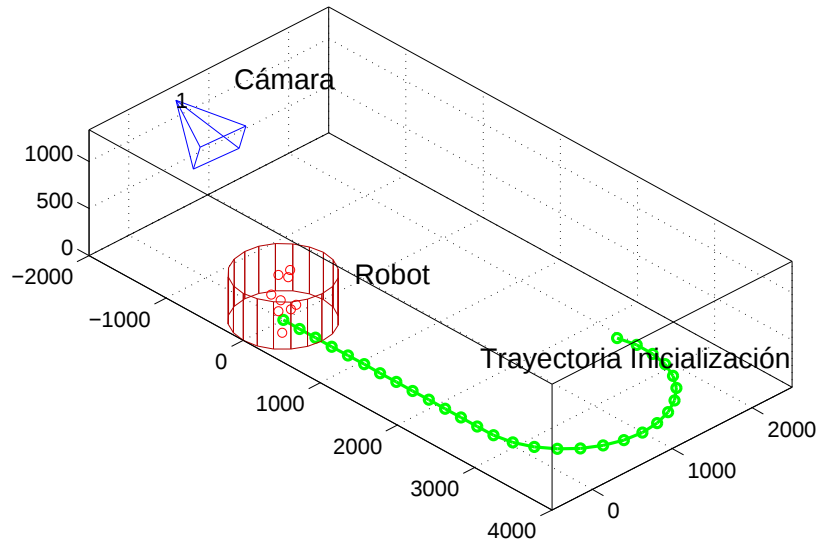


Figura 5.9: Configuración geométrica de la escena utilizada para el experimento de inicialización.

donde T_0 y α_0 representan el valor del vector de posición y el ángulo de orientación del robot respectivamente.

- Los experimentos se realizan para todas las posibilidades de función de coste planteadas, describiendo cada una mediante los acrónimos siguientes:
 1. Función de coste con matriz de covarianza Σ_L completa (M.C.).
 2. Función de coste con matriz de covarianza Σ_L aproximada por bloques (M.B.).
 3. Función de coste con matriz de covarianza Σ_L aproximada mediante bloques 2×2 (M.D.).
 4. Función de coste con matriz de covarianza Σ_L igual a la matriz identidad (M.I.).
- En todos los experimentos realizados se ha utilizado el método de inicialización previo a la optimización que se ha presentado en este capítulo para obtener un valor de inicial en el vector de parámetros.

Bajo las condiciones anteriormente expuestas, se realizan los siguientes experimentos, en los cuales se mostrarán los resultados para las cuatro posibilidades de función de coste planteadas:

- Comparación del error de inicialización ante variaciones en el error que se comete en odometría. Para ello se variará el valor de Σ_W mediante un factor multiplicativo ρ

que variará en el intervalo:

$$\rho \in (0,01 \quad 0,1 \quad 0,5 \quad 1 \quad 5 \quad 10)$$

Para este experimento se supondrá que $p_{out} = 0$. En las figuras 5.10.a, 5.10.b y 5.10.c se muestra el error de inicialización en función de la variación del parámetro ρ .

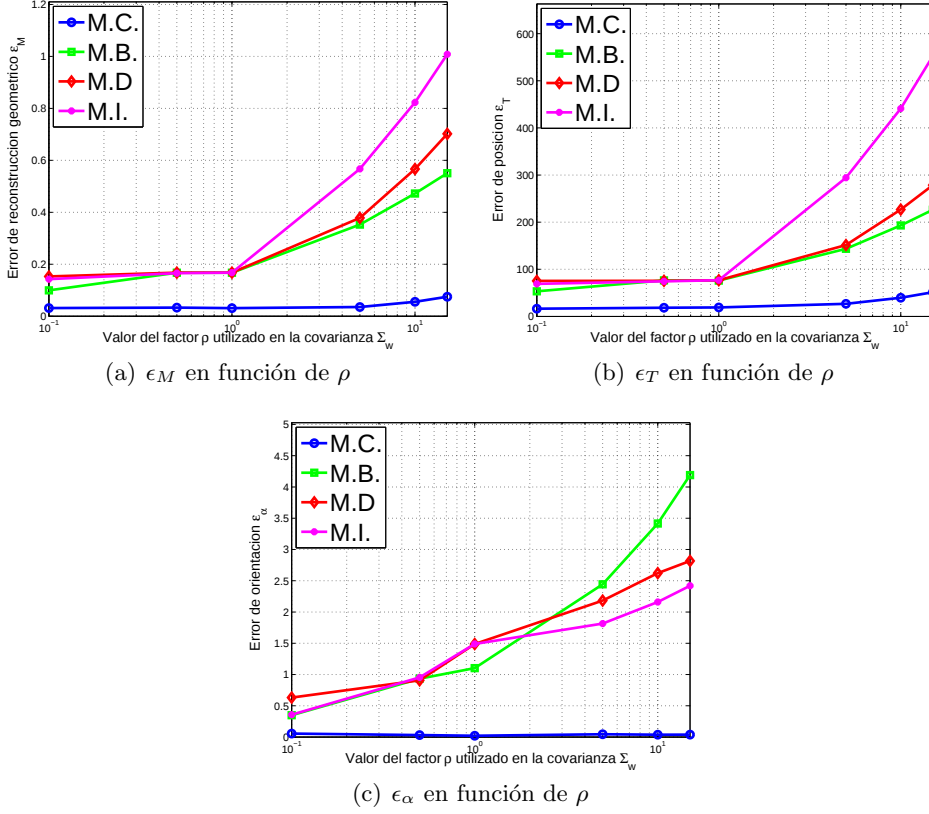


Figura 5.10: Experimento que muestra la variación del error de inicialización en función del error en la odometría.

Como se muestra en la figura 5.10, el aumento en el ruido presente en la odometría, única fuente métrica en el proceso, afecta con diverso impacto en los errores de inicialización de los diversos métodos presentados. El método que considera la matriz Σ_L como una matriz completa o M.C. mantiene un nivel de error similar ante las variaciones en el error de la odometría, lo cual indica la capacidad del modelo de máxima verosimilitud de desacoplar el error en la odometría del error de reconstrucción. El resto de aproximaciones sigue una evolución diferenciada, en la que el error de odometría eleva el error de la reconstrucción considerablemente, tanto en pose como en geometría. A pesar de ello se puede observar como los métodos M.B. y M.D. mejoran el método que aproxima Σ_L como una matriz identidad, con el que se obtiene una función de coste convencional. Como regla general, a medida que se aproxima la función de coste a aquella resultado del enfoque de máxima verosimilitud, el cual se modela con el método M.C., más capacidad posee el método propuesto de eliminar la influencia del ruido de la odometría de la solución para la pose y geometría.

- Comparación del error de inicialización en función de la longitud total de la trayectoria. Mediante un factor de escala ρ_t se realiza un escalado isotrópico de la trayectoria utilizada para la inicialización, de forma que se pueda comprobar la relación entre la longitud de la trayectoria y el error de inicialización. El parámetro de escala se varía en el rango siguiente:

$$\rho_t = (0,2 \quad 0,4 \quad 0,6 \quad 0,8 \quad 1 \quad 1,2 \quad 1,4)$$

En la figura 5.11, se muestra el error de inicialización en función de la longitud de la trayectoria.

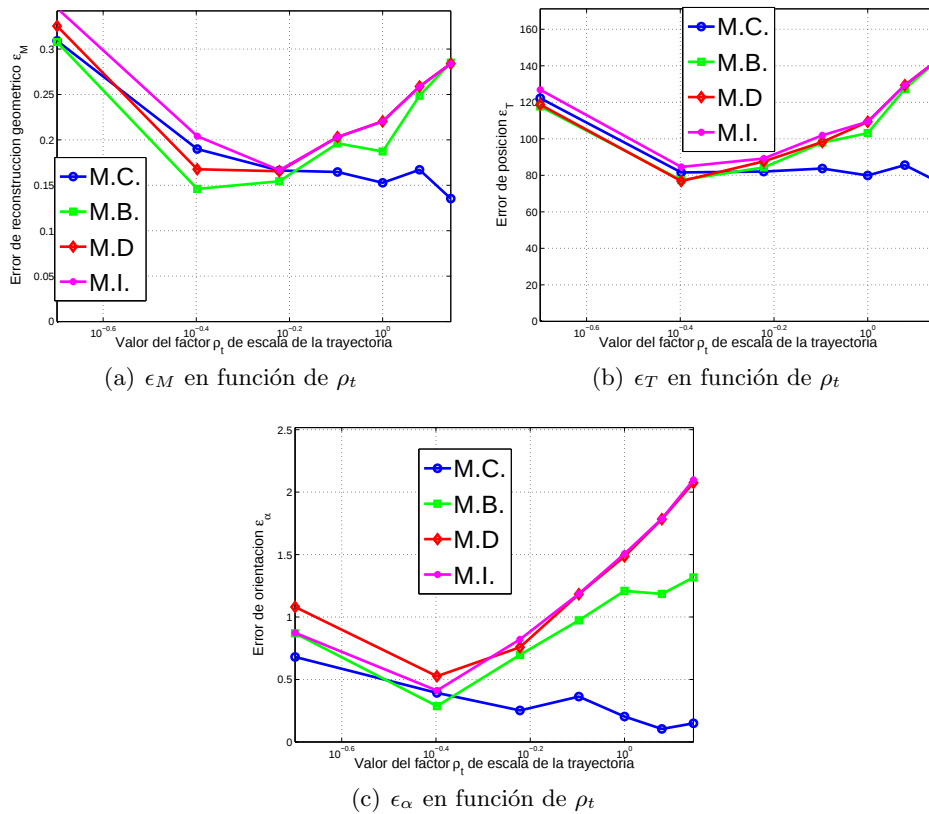


Figura 5.11: Experimento que muestra la variación del error de inicialización en función de la escala aplicada a la trayectoria aplicada a la inicialización.

La figura 5.11 pone de manifiesto que, debido al ruido acumulativo de la odometría, un aumento de la trayectoria utilizada en la inicialización supone un incremento notable en los errores de inicialización en los métodos M.B, M.D y M.I. Sin embargo se observa como el método M.C. es capaz de eliminar eficazmente el error acumulativo presente en las medidas de odometría de la reconstrucción. Por otro lado, si la trayectoria es reducida, el error también aumenta, debido principalmente al error de medida y la disminución en la calidad de la información. Este hecho produce que exista un tamaño óptimo de trayectoria para los métodos propuestos a excepción del método M.C., el cual es capaz de mantener el error con la longitud de la trayectoria.

- Comparación del error de inicialización en función de la cantidad de “outliers” contenidos en las medidas. Se varía el valor de p_{out} en el rango:

$$p_{out} \in (0 \ 0,05 \ 0,1 \ 0,15 \ 0,2),$$

mostrándose los resultados de aplicar el método de inicialización con función robusta en las figuras 5.12.a, 5.12.c y 5.12.e.

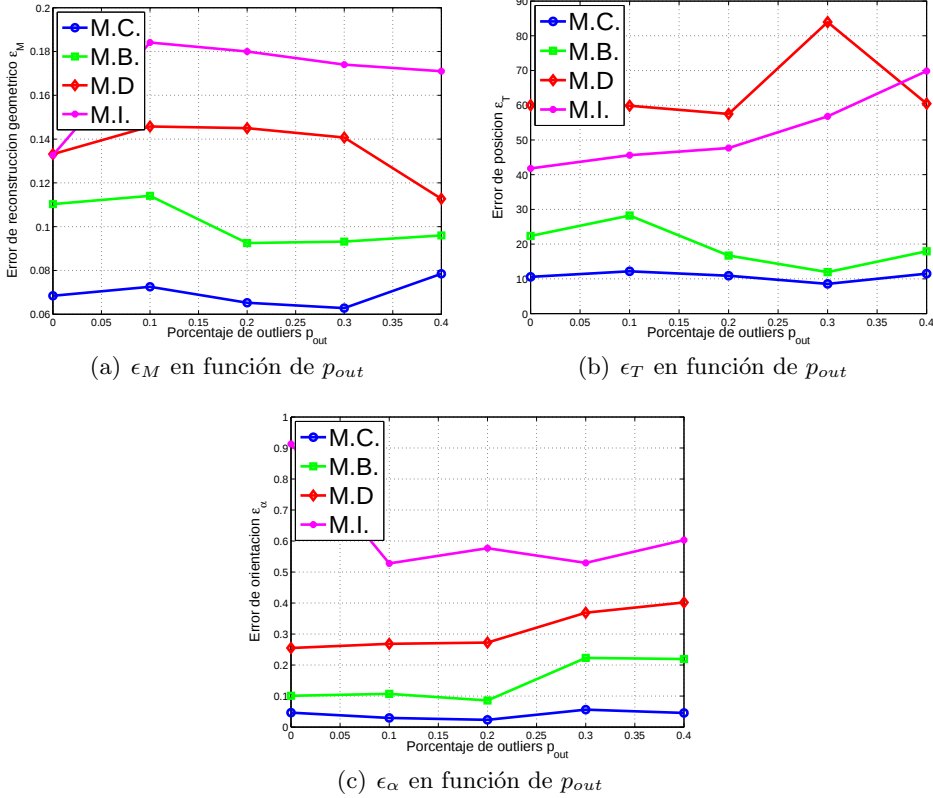


Figura 5.12: Experimento que muestra la variación del error de inicialización en función del porcentaje de “outliers” en la medida.

5.3. Inicialización de la pose y geometría del robot mediante múltiples cámaras

En esta sección se aborda el problema de la obtención de la posición y geometría inicial del robot visto por un mínimo de dos cámaras. Debido a la estructura de cámaras alejadas, donde cada cámara es susceptible de ofrecer una instantánea complementaria de la geometría del robot, se propone en esta sección un enfoque de reconstrucción a partir de movimiento, similar a la utilizada en el caso monocular, en la que no sean estrictamente necesarias las correspondencias entre las diferentes cámaras. La idea básica que se plantea en esta tesis es que cada cámara obtenga de manera independiente una serie de “tracks”

o trayectorias correspondientes a puntos de la geometría del robot en movimiento durante una secuencia de inicialización. La información de cada cámara se combina para ofrecer una solución única a la pose del robot y la reconstrucción de su geometría, vista por todas las cámaras. Bajo este enfoque, los procesos de emparejamiento de puntos entre cámaras separadas no son necesarios y son reemplazados por procesos de seguimiento, que en general representan un problema más sencillo y mejor condicionado.

Al igual que en el caso de una cámara, se propondrá el uso de una técnica de “Bundle Adjustment”, que es capaz de reconstruir eficientemente la posición del robot y su geometría, utilizando para ello movimiento. Sin embargo, en este caso, la odometría no es estrictamente necesaria, ya que las restricciones geométricas impuestas por la calibración de las cámaras son suficientes para obtener la pose del robot a lo largo de la trayectoria y su estructura.

En la figura 5.13 se resume el proceso de inicialización de la pose y geometría del robot móvil utilizando múltiples cámaras.

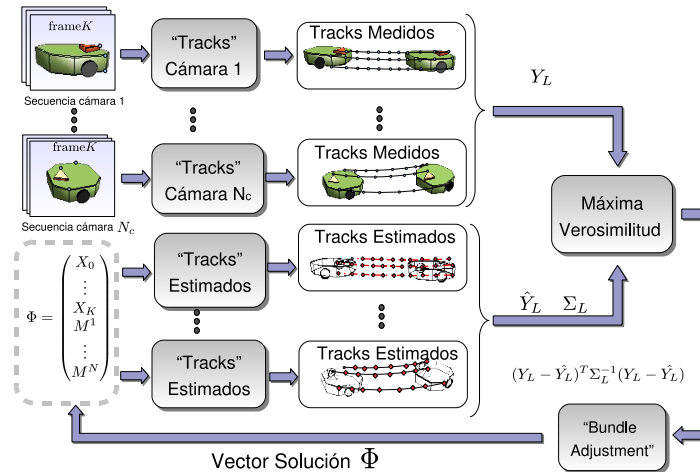


Figura 5.13: Estructura del proceso de inicialización mediante múltiples cámaras

5.3.1. Definiciones preliminares

La secuencia de inicialización consta, del mismo modo que en el caso de una sola cámara, de $K + 1$ instantes de tiempo que comienzan con $k = 0$, en los cuales el robot está en movimiento. Los vectores $X_0, \dots, X_K$ definen la posición y orientación del robot en cada instante de tiempo de la trayectoria. A pesar de que la odometría no es estrictamente necesaria, siempre que exista y partiendo del valor de X_0 se puede obtener una estimación susceptible de contener ruido acumulativo utilizando el conjunto de lecturas de odometría $U_1, \dots, U_K$ y el modelo de movimiento recursivo g .

Se supone la existencia de un conjunto de $N_c > 2$ cámaras involucradas en el proceso de inicialización que, por motivos de claridad, suponemos que observan la geometría del robot en la secuencia completa. En cada cámara, indexada con el índice n , se realiza el

seguimiento de un conjunto de N_n puntos pertenecientes a la estructura del robot. No se puede asegurar a priori que en cada cámara se detecten el mismo número de puntos ni puntos coincidentes, por lo que N_n es necesariamente dependiente del índice de la cámara.

En el frame k de la secuencia y suponiendo la cámara n , el vector de medidas consiste en lo siguiente:

$$Y_k^n = \begin{pmatrix} y_k^{1,n} & \cdots & y_k^{N_n,n} \end{pmatrix} \quad y_k^{i,n} = (u_k^{i,n}, v_k^{i,n}), \quad (5.94)$$

El vector de medidas global, que en este caso se representa directamente mediante el vector Y_L , se forma mediante la concatenación en una misma columna de todas las medidas, desde todas las cámaras y para todos los instantes de tiempo que componen la trayectoria.

$$Y_L = \left(\underbrace{(Y_1^1 \cdots Y_K^1)}_{\text{Cámara 1}} \cdots \underbrace{(Y_1^{N_c} \cdots Y_K^{N_c})}_{\text{Cámara } n} \right)^T \quad (5.95)$$

De este modo, el conjunto $y_0^{i,n}, \dots, y_K^{i,n}$ describe la posición de un mismo punto en los distintos instantes de tiempo, y el conjunto $Y_0^n, \dots, Y_K^n$, representa el conjunto de medidas completo de la cámara n .

Cada medida individual $y_k^{i,n}$ es función de los parámetros de la cámara n a la que pertenece, el vector de pose X_k y el punto de la estructura del robot correspondiente:

$$y_k^{i,n} = h(t(X_k, M^j), P^n) + \mathbf{v}_k^{i,n} \quad \mathbf{v}_k^{i,n} = N(0, \Sigma_v), \quad (5.96)$$

donde la correspondencia de M^j con el punto i y la cámara n es dado por el ordenamiento correspondiente en el vector de estados:

$$X_k^a = \left(X_k \quad \underbrace{(M^1 \cdots M^{N_1})}_{\text{Cámara 1}} \cdots \underbrace{(M^r \cdots M^{r+N_{N_c}})}_{\text{Cámara } N_c} \right) \quad r = \sum_{n=1}^{N_c-1} N_n. \quad (5.97)$$

El vector aumentado X_k^a aglutina la geometría del robot vista por cada cámara, aunque referida siempre en el mismo sistema de coordenadas, utilizado para definir la pose del robot. Por el momento se supone que las cámaras observan puntos diferentes del robot y por tanto no se considera el caso de que la expresión (5.97) contenga repeticiones del mismo punto visto por diferentes cámaras.

5.3.2. Función de coste

El conjunto completo de incógnitas se compone del conjunto de poses en la trayectoria $X_0, \dots, X_K$ y las coordenadas tridimensionales de puntos vistos por las cámaras $M^1, \dots, M^N$, donde $N = \sum_{n=1}^{N_c} N_n$. Todos los parámetros de interés se agrupan en el vector Φ , correspondiente a X_0^a y el resto de poses de la trayectoria:

$$\Phi = (X_0, \cdot, X_K, M^1, \dots, M^N) \quad N = \sum_{n=1}^{N_c} N_n \quad (5.98)$$

Se plantea un enfoque de máxima verosimilitud, como el planteado en el caso monocular, donde el objetivo se centra en la obtención del valor de Φ que maximiza la función de densidad de probabilidad $p(Y_L|\Phi)$:

$$\max_{X_0^a} p(Y_L|\Phi) = \max_{\Phi} \left\{ \frac{1}{2^{N_L/2} |\Sigma_L|} e^{-\frac{1}{2}(Y_L - \hat{Y}_L)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L)} \right\}, \quad (5.99)$$

donde Y_L representa una realización concreta del proceso $\mathbf{Y}_L$ determinado por la f.d.p $p(Y_L|\Phi)$, es decir, el conjunto de medidas tomadas por el conjunto de cámaras a lo largo de la trayectoria de inicialización.

Tomando el logaritmo negativo de la expresión (5.99) se obtiene un problema equivalente en el que se busca el mínimo en función de X_0^a de la siguiente función de coste:

$$\min_{\Phi} (Y_L - \hat{Y}_L)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L), \quad (5.100)$$

La expresión (5.100) consiste en una función de coste cuadrática que en función de Φ compara la distancia de Mahalanobis entre las medidas reales Y_L y aquellas que son estimadas a partir de Φ o $\hat{Y}_L$, y que son obtenidas a partir de los modelos de proyección en la cámara.

La matriz Σ_L representa la matriz de covarianza resultante del vector Y_L , aunque en esta ocasión se trata de una matriz diagonal por bloques, donde cada bloque consiste en el ruido atribuido a cada medida Σ_v . La simplicidad de Σ_L en este caso contrasta con las diferentes alternativas que se propusieron en el problema de una sola cámara. En este caso, puesto que la pose completa en cada instante de tiempo es un parámetro independiente del vector objetivo Φ , las relaciones estadísticas temporales entre las distintas medidas desaparecen, lo que simplifica enormemente la función de coste, permitiendo reescribirla en la siguiente adición simple de términos:

$$\epsilon^2 = \sum_{k=1}^K \sum_{n=1}^{N_c} \sum_{i=1}^{N_n}, (y_k^i - \hat{y}_k^i) \Sigma_v^{-1} (y_k^i - \hat{y}_k^i)^T \quad (5.101)$$

,

La obtención del mínimo de (5.101) con respecto a Φ representa la reconstrucción completa de la trayectoria y la geometría del robot. Con el objetivo de alcanzar el mínimo de la función, se propone el mismo método Levenberg-Marquardt, utilizado en el caso monocular. Del mismo modo, las derivadas analíticas de las funciones involucradas en el cálculo de la función de coste son imprescindibles en este caso. Una breve descripción del algoritmo se presenta en el algoritmo 6.

Algoritmo 6 Obtención del mínimo de (5.101) mediante Levenberg-Marquardt

-
- ```

1: Requerimientos:
2: Medidas en el plano imagen Y_L
3: Parámetros para los modelos de proyección h
4: Valor inicial para X_0^a , $numiter_{max}$ y ϵ_{min}
5:
6: while $\epsilon > \epsilon_{min}$ and $numiter < numiter_{max}$ do
7: Obtener $\hat{Y}_L$ y Σ_L usando (5.95).
8: Cálculo de J y H .

```

$$J = -2\left(\frac{\partial \hat{Y}_L}{\partial X_0^a}\right)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L) \quad (5.102)$$

$$H \approx 2\left(\frac{\partial \hat{Y}_L}{\partial X_0^a}\right)^T \Sigma_L^{-1} \left(\frac{\partial \hat{Y}_L}{\partial X_0^a}\right) \quad (5.103)$$

- ```

9:    $\epsilon_{LM} = \infty$ 
10:  while  $\epsilon_{LM} > \epsilon$  do
11:     $H_{LM} = H + \delta I$ 
12:     $\Delta_{X^a} = -(H)^{-1} J$ 
13:     $X_{LM}^a = X_0^a + \Delta_{X^a}$ 
14:     $\epsilon_{LM} = (Y_L - \hat{Y}_L)^T \Sigma_L^{-1} (Y_L - \hat{Y}_L)$ 
15:     $\delta = \delta \cdot 10$ 
16:  end while
17:   $\epsilon = \epsilon_{LM}$ 
18:   $\delta = \delta/100$      $X_0^a = X_{LM}^a$ 
19: end while

```
-

Mecanismo de agregación

El conjunto de puntos $M^1, \dots, M^N$, contenidos en el vector Φ y resultado de la minimización de (5.101), está formado por la aglutinación en el mismo vector de cada porción de la geometría vista en cada cámara. Dicha geometría se supone complementaria, sin embargo, en la práctica existirá un cierto nivel de solape entre los puntos vistos por diferentes cámaras. En este punto se propone un método que refina la solución Φ de tal manera que los puntos coincidentes en varias cámaras sean tratados como tal y por tanto se eliminen las versiones repetidas de los mismos. A su vez, el hecho de identificar dichos puntos es de interés, puesto que permite recalcular el vector entero con mayor precisión.

Se propone un método sencillo para la identificación de los puntos repetidos. Una vez obtenida la inicialización, y si ésta finalizó con éxito mediante un error de reconstrucción bajo, se realiza la suposición de que aquellos puntos M^q en la cámara n_p , y M^p en la cámara n_q , incluidos en Φ que representan el mismo punto desde diferentes cámaras obtengan valores muy próximos en sus coordenadas reconstruidas. Se impone un criterio de distancia euclídea $\|M_p - M_q\| < d_{\min}$ entre cada par de candidatos a ser un punto coincidente. De esta manera, se considera que dos puntos suficientemente cerca son identificados como uno sólo.

Una vez identificadas las coincidencias del vector Φ , se repite el mecanismo de optimización teniendo en cuenta este hecho. La única modificación que requiere el algoritmo es la eliminación de todas las repeticiones menos una de los puntos identificados. Dentro de la

optimización el Hessiano y el Jacobiano deberán tener en cuenta que puntos del plano imagen de las cámaras involucradas son en realidad dependientes funcionalmente de uno solo. De esta manera, la optimización se lleva a cabo teniendo en cuenta las nuevas restricciones utilizadas.

5.3.3. Tratamiento de los “outliers”

De un modo análogo al caso monocular y asumiendo la existencia de puntos en Y_L considerados “outliers”, la función de coste mostrada en (5.101) es modificada para incluir una función robusta. Puesto que la matriz Σ_L produce residuos separables por cada medida independiente, la aplicación de la función de coste se puede expresar como:

$$\epsilon^2 = \sum_{k=1}^K \sum_{n=1}^{N_c} \sum_{i=1}^{N_n} \rho((y_k^i - \hat{y}_k^i) \Sigma_v^{-1} (y_k^i - \hat{y}_k^i)^T), \quad (5.104)$$

siendo $\rho(\epsilon^2)$, la función mostrada en (5.49). El uso de funciones robustas, como ya fue comentado con anterioridad, afectan a la convergencia del algoritmo, por lo que deberá existir un método para inicializar el valor de Φ en la primera iteración.

5.3.4. Tiempo de cómputo

El tiempo de cómputo necesario para llegar a una solución para Φ requiere de un conjunto de iteraciones del algoritmo de optimización Levenberg-Marquardt. Como se comentó con anterioridad, si la aproximación cuadrática de la función es suficientemente precisa, la precisión aumenta en dos decimales con cada iteración, por lo que la convergencia esperada en tal caso debería ser rápida aunque dependiente de muchos factores de difícil análisis, como pueden ser los mínimos locales de la función o la eficiencia del método de obtención de la solución inicial que se comentará más adelante en este capítulo.

La complejidad algorítmica para resolver cada iteración del algoritmo 6 se debe al cálculo de las derivadas de las funciones de proyección y en gran parte en la obtención de la solución del sistema $H\Delta_{X^a} = -J$.

La estructura que presenta la matriz Hessiana se muestra en la figura 5.14 para un ejemplo ilustrativo. En todo caso se puede observar una estructura de tipo disperso (“flecha”) y por lo tanto se puede acudir a métodos que aprovechan esta característica para acelerar el cálculo de la solución Δ_{X^a} . En la presente tesis se propone el uso del método de “Schur” para resolver el problema mediante un método con complejidad $\mathcal{O}(N^2)$.

En el caso de utilizar funciones robustas, la matriz Hessiana tiene la misma estructura, ya que la función ρ es aplicada individualmente a cada bloque 2×2 . El método robusto tendrá impacto en el número de iteraciones necesarias para llegar al mínimo y en el cálculo de las derivadas.

$$H = \begin{pmatrix} \begin{matrix} \text{[Sparse Block]} & \text{[Dense Block]} \\ \hline \text{[Dense Block]} & \text{[Sparse Block]} \end{matrix} \\ \begin{matrix} X_0, \dots, X_K \\ M^1, \dots, M^N \end{matrix} \end{pmatrix}$$

Figura 5.14: Matriz Hessiana para $K = 14$, $N_c = 3$ y $N = 6$

5.3.5. Inicialización de la solución previa a la optimización

El método propuesto para obtener el mínimo de la función de coste requiere un valor de inicialización para el vector Φ con el que comenzar a iterar. Se proponen dos metodologías diferentes en función de la información presente en el sistema.

- **Disponibilidad de lecturas de odometría del robot:** en este caso se propone un algoritmo de reconstrucción en el que el origen de coordenadas del robot queda fijado por la odometría, obteniendo un algoritmo similar al utilizado en el caso de una sola cámara, donde se obtenía una solución lineal.
- **Robot sin sistema de odometría:** se propone un algoritmo sencillo que obtenga una solución inicial para el caso de no utilizar la odometría del robot. El principal interés de este caso radica en la utilización del sistema de posicionamiento para objetos o robots no controlados por el espacio inteligente.

Iniciación utilizando varias cámaras y odometría

Se plantea en este punto el uso de la información proveniente de la odometría presente en el robot para obtener una solución inicial para el vector de incógnitas necesario para la inicialización. El uso de la odometría permite obtener una solución no iterativa y además la posición del origen de rotación del robot, de tal modo que es posible utilizar la odometría como estimación de movimiento en el método secuencial.

En este apartado se expone el problema desde un enfoque similar al utilizado en la sección 5.2 en la que se obtenía un resultado homólogo utilizando una sola cámara.

Sea un punto M^i perteneciente a la estructura del robot con una pose X_k , entonces su proyección en el plano imagen de la cámara n , $y_k^{i,n}$ será:

$$\begin{pmatrix} y_k^{i,n} \\ 1 \end{pmatrix} = \lambda K_c^n (R_c^n (R_k M^i + T_k) + T_c^n), \quad (5.105)$$

donde λ representa el factor de escala de la transformación no homogénea y R_c^n, T_c^n, K_c^n corresponden a los parámetros de calibración de la cámara n , en la que el punto i es detectado. La matriz de rotación R_k y el vector de traslación T_k representan la pose del robot en el instante k en forma de transformación afín. Su relación con las variables del vector de estado $X_k = (x_k^1, x_k^2, \alpha_k)$, es idéntica a la expresada en (5.55).

Utilizando la expresión (5.57), se establece una relación entre la pose en el instante k y la pose inicial descrita por R_0 y T_0 :

$$T_0 = \begin{pmatrix} x_0^1 \\ x_0^2 \\ 0 \end{pmatrix} \quad R_0 = \begin{pmatrix} \cos(\alpha_0) & -\sin(\alpha_0) & 0 \\ \sin(\alpha_0) & \cos(\alpha_0) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.106)$$

La expresión (5.106) en función de R_0 y T_0 será entonces:

$$\begin{pmatrix} y_k^i \\ 1 \end{pmatrix} = \lambda K_c^i (R_c^i (R_k^\Delta R_0 M^i + T_0 + R_0 T_\Delta) + T_c^i). \quad (5.107)$$

Al igual que ocurría en el caso monocular, el hecho de considerar funciones trigonométricas en el ángulo y la necesidad de calcular el valor de lambda, el cual depende de la inversa de muchos parámetros, impide abordar el problema directamente.

Mediante el cambio de parámetros $a = \cos(\alpha_0)$, $b = \sin(\alpha_0)$, y suponiendo que $M_0^i = R_0 M^i$ se obtiene un nuevo vector de incógnitas Φ idéntico al expresado en (5.61):

$$\Phi = (x_0^1 \quad x_0^2 \quad a \quad b \quad (M_0^1)^T, \dots, (M_0^N)^T)^T \quad (5.108)$$

Desglosando en forma vectorial la transformación proyectiva se obtiene:

$$\begin{pmatrix} u_k^{i,n} \\ v_k^{i,n} \end{pmatrix} = y_k^{i,n} \quad \begin{pmatrix} U_k^{i,n} \\ V_k^{i,n} \\ S_k^{i,n} \end{pmatrix} = K_c^n (R_c^n (R_k^\Delta M_0^i + T_0 + L_{T_k^\Delta}) + T_c^n), \quad (5.109)$$

donde gracias a la nueva parametrización las componentes $U_k^{i,n}$, $V_k^{i,n}$ y $S_k^{i,n}$ equivalen a una transformación lineal de los parámetros Φ :

$$U_k^{i,n} = L_{U_k}^{i,n} \Phi + b_{U_k}^{i,n} \quad V_k^{i,n} = L_{V_k}^{i,n} \Phi + b_{V_k}^{i,n} \quad S_k^{i,n} = L_{S_k}^{i,n} \Phi + b_{S_k}^{i,n} \quad (5.110)$$

Haciendo uso de las propiedades del producto vectorial y para evitar la aparición del parámetro λ , se obtienen dos relaciones linealmente independientes del vector Φ por cada medida $y_k^{i,n}$:

$$\begin{cases} v_k^{i,n} S_k^{i,n} - V_k^{i,n} = 0 \\ -u_k^{i,n} S_k^{i,n} + U_k^{i,n} = 0 \end{cases} \rightarrow \begin{cases} (v_k^i L_{S_k}^{i,n} - L_{V_k}^{i,n}) \Phi = v_k^{i,n} b_{S_k}^{i,n} - b_{V_k}^{i,n} \\ (-u_k^{i,n} L_{S_k}^{i,n} + L_{U_k}^{i,n}) \Phi = -u_k^{i,n} b_{S_k}^{i,n} + b_{U_k}^{i,n} \end{cases} \quad (5.111)$$

Utilizando todas las medidas tomadas por todas las cámaras en la trayectoria de inicialización $k = 1, \dots, K$, se obtiene un sistema lineal, en general sobredeterminado, para el vector de incógnitas Φ .

$$A\Phi = B \quad A = \begin{pmatrix} A_1 \\ \vdots \\ A_{N_c} \end{pmatrix} \quad B = \begin{pmatrix} B_1 \\ \vdots \\ B_{N_c} \end{pmatrix}, \quad (5.112)$$

donde la principal diferencia con respecto al caso monocular radica en el hecho de que la matriz A se puede dividir en bloques correspondientes a medidas sólo determinadas por una cámara concreta con parámetros de calibración determinados.

$$A_n = \begin{pmatrix} (v_1^{1,n} L_{S_1}^{1,n} - L_{V_1}^{1,n}) \\ (-u_1^{1,n} L_{S_1}^{1,n} + L_{U_1}^{1,n}) \\ \vdots \\ (v_K^{N,n} L_{S_K}^{N,n} - L_{V_K}^{N,n}) \\ (-u_K^{N,n} L_{S_K}^{N,n} + L_{U_K}^{N,n}) \end{pmatrix} \quad B_n = \begin{pmatrix} v_1^{1,n} b_{S_1}^{1,n} - b_{V_1}^{1,n} \\ -u_1^{1,n} b_{S_1}^{1,n} + b_{U_1}^{1,n} \\ \vdots \\ v_K^{N,n} b_{S_K}^{N,n} - b_{V_K}^{N,n} \\ -u_K^{N,n} b_{S_K}^{N,n} + b_{U_K}^{N,n} \end{pmatrix} \quad (5.113)$$

En el caso de múltiples cámaras, el sistema no contiene ningún valor singular nulo, y por tanto mediante el uso de la pseudoinversa de A se obtiene directamente una estimación del vector Φ :

$$\Phi = (A^T A)^{-1} A^T B \quad (5.114)$$

Debido al ruido presente en las medidas, es probable que el valor de a y b no cumplan exactamente la condición de contorno $a^2 + b^2 = 1$. Con el objetivo de encontrar los valores de $a + \Delta_a$ y $b + \Delta_b$ que cumplen la condición de contorno, se obtiene la descomposición en valores singulares de la matriz R_0 que resultaría de utilizar a y b :

$$R_0 = \begin{pmatrix} a & b & 0 \\ -b & a & 0 \\ 0 & 0 & 1 \end{pmatrix} = U \begin{pmatrix} \alpha_1 & 0 & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 \end{pmatrix} V^T \quad (5.115)$$

Si los valores de a y b fueran los correctos, se cumple que $\alpha_1 = \alpha_2 = \alpha_3 = 1$. En general, una aproximación de R_0 se consigue forzando dichos valores a la unidad:

$$R_0 = UV^T \quad \Delta_a = R_0(1,1) - a \quad \Delta_b = R_0(2,2) - b \quad (5.116)$$

La solución buscada del vector X_0^a requiere invertir la parametrización utilizada:

$$\alpha_0 = \tan^{-1}(a + \Delta_a, b + \Delta_b) \quad M^i = R_0^{-1} M_0^i \quad X_0^a = (X_1 \ \cdots \ X_K \ (M^1)^T \ \cdots \ (M^N)^T)^T, \quad (5.117)$$

donde la pose del robot en la secuencia $X_1, \dots, X_K$ es obtenida mediante el uso de la estimación de la odometría a partir de la pose inicial $X_0 = (x_0^1, x_0^2, \alpha_0)$

Inicialización de Φ utilizando varias cámaras

Para este caso se propone un método muy simple que permite obtener una aproximación inicial a la solución Φ . Consiste en los siguientes pasos:

- Para cada instante de tiempo k y cada cámara n , se obtiene el centro de masas μ_k^n de los puntos incluidos en Y_k^n :

$$\mu_k^n = \frac{1}{N_n} \sum_{i=1}^{N_n} y_k^i \quad (5.118)$$

- La posición del robot, excluyendo la orientación, p_k se obtiene mediante la suposición de que el conjunto de centros de gravedad en cada cámara corresponden al mismo punto en el espacio, lo que permite utilizar una función de triangulación:

$$p_k = Tr(\mu_k^1, \dots, \mu_k^{N_c}, P_1, \dots, P_{N_c}) \quad (5.119)$$

- Una vez que las $K + 1$ posiciones del robot $p_0, \dots, p_K$ se han obtenido mediante triangulación de los centros de masas, el conjunto de orientaciones $\alpha_0, \dots, \alpha_K$ se obtienen de manera que se ajusten al tipo de movimiento que se espera en el objeto a detectar. En general los objetos en movimiento presentan algún tipo de movimiento no holonómico, de tal manera que su orientación está restringida de alguna forma a la propia curvatura de la trayectoria realizada. En tal caso, la orientación del objeto se calcula fácilmente a partir de las posiciones $p_0, \dots, p_K$. Sin embargo, en aquellos casos en los que el movimiento del objeto sea enteramente holonómico esta suposición deja de ser cierta. En dichos casos se puede utilizar un valor aleatorio.

Una vez obtenido el conjunto de orientaciones, la pose del robot en la trayectoria de inicialización $X_0, \dots, X_K$ se define concatenando la posición y su respectiva orientación:

$$X_k = (p_k, \alpha_k) \quad k = 0, \dots, K \quad (5.120)$$

- La geometría del robot $M^1, \dots, M^N$ se inicializa de manera aleatoria alrededor del volumen ocupado por una esfera de radio R_e y centrada en el origen. El radio de la esfera se calcula de tal manera que sea el mínimo radio que contiene a las medidas de todas las cámaras en un frame específico.

5.3.6. Obtención de la distribución gaussiana de la solución

Una vez obtenido el mínimo de (5.101), es necesaria la obtención de la matriz de covarianza que representa el vector aumentado X_0^a , en el instante inicial. De esta manera se hace posible conectar la inicialización con el algoritmo online que se trata en el capítulo 7.

Al igual que se propone para una cámara, la covarianza del vector Φ se obtiene fácilmente a partir de las derivadas de primer orden de los residuos alrededor del mínimo obtenido. De esta manera Σ_Φ se obtiene utilizando la siguiente expresión:

$$\Sigma_\Phi = (J_\Phi^T \Sigma_\Phi^{-1} J_\Phi)^{-1}, \quad (5.121)$$

donde J_Φ es el Jacobiano de $\hat{Y}$ con respecto Φ usado en la optimización. La matriz de covarianza del vector aumentado X_0^a y representada como Σ_0^a , se obtiene fácilmente mediante truncamiento de Σ_Φ .

5.3.7. Configuraciones degeneradas

Las configuraciones degeneradas en el caso de múltiples cámaras son mucho menos restrictivas que aquellas resultado del uso de una sola cámara en combinación con la odometría. En este caso, la trayectoria que sigue el robot puede ser circular, o rectilínea. Existen no obstante ciertas configuraciones de múltiples cámaras y geometría vista por ellas, que da lugar a reconstrucciones ambiguas [Hartley and Zisserman, 2003]. El sistema propuesto tiene una baja probabilidad de dar con una de dichas configuraciones, y por tanto el problema no se considera en esta tesis.

5.3.8. Evaluación experimental del método propuesto

A continuación se presenta un conjunto de experimentos realizados en un entorno simulado, que al igual que en el caso de la inicialización con una cámara, tiene como objetivo evaluar el rendimiento del algoritmo propuesto ante la variación de diferentes parámetros.

El entorno de prueba elegido para esta simulación consiste en un total de $N_c = 8$ cámaras, dispuestas como se muestra en la figura 5.15. La distribución coincide con una distribución real de “Espacio Inteligente”, en la que se mostrarán resultados reales en el capítulo 8. Las características del entorno de prueba se resumen a continuación:

- Se dispone de $N_c = 8$ cámaras “pin-hole”, con parámetros intrínsecos

$$(\alpha_u = 636,7888, \alpha_v = 637,5610, u_0 = 313,3236, v_0 = 210,6894),$$

y la posición y orientación con respecto al origen del mundo O_W que se muestra en la figura 5.15. La matriz Σ_V que modela el ruido de medida, se supone diagonal con un valor de $\sigma_v^2 = 10$ en su diagonal principal.

- El robot consta de un conjunto de $N = 6N_c$ puntos distribuidos aleatoriamente dentro de un cilindro de radio $R = 0,5m$ y altura $h = 1m$, lo cual corresponde con la

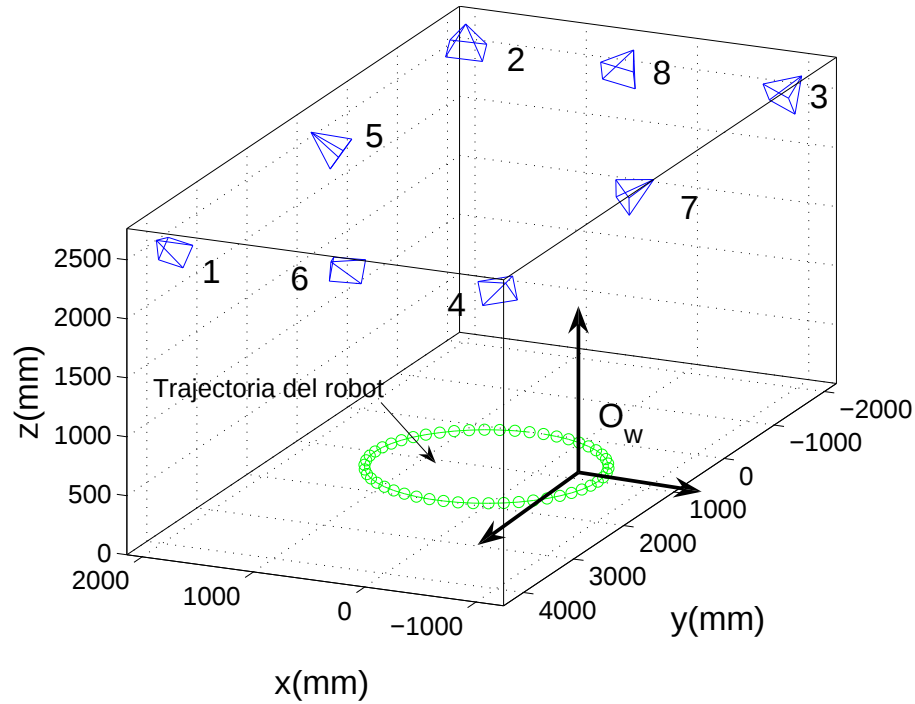


Figura 5.15: Configuración geométrica de la escena utilizada para el experimento de inicialización mediante múltiples cámaras.

altura máxima que tendrá el tipo de robot utilizado. Cada cámara observa un total de $N_c = 6$ puntos diferentes de la estructura del robot y se supone que no existe solapamiento.

- Se ha simulado un sistema de odometría en el robot en el que la incertidumbre en las lecturas de odometría se representa mediante la matriz diagonal Σ_W , definida por las varianzas del ruido de la lectura de velocidad lineal y angular $\sigma_{v_l}^2 = 10$ y $\sigma_{\Omega}^2 = 1$ respectivamente.
- La trayectoria de inicialización utilizada en los experimentos se visualiza en la figura 5.15, y consta de una circunferencia de $1m$ de radio.
- La generación de “outliers” utiliza el mismo sistema y parámetros presentados en los experimentos de una sola cámara que se expusieron en el punto 5.2.10.
- Los errores en la inicialización se dividen, al igual que en el punto 5.2.10, en tres magnitudes, ϵ_M , ϵ_T y ϵ_α , que describen el error en la geometría del robot, posición y orientación respectivamente.

Los experimentos realizados, bajo las condiciones anteriormente citadas, se describen a continuación:

- Comparación del error de inicialización en función del de la cantidad de “outliers”

contenidos en las medidas. Se varía el valor de p_{out} en el rango:

$$p_{out} \in (0 \quad 0,05 \quad 0,1 \quad 0,15 \quad 0,2),$$

mostrándose los resultados de aplicar el método de inicialización con función robusta en las figuras 5.16.b, 5.16.d y 5.16.f y en ausencia de ella en las figuras 5.16.a, 5.16.c y 5.16.e.

Como puede observarse en las figuras que están en la columna izquierda de la figura 5.16, en la que se muestran los errores mediante un método de inicialización no robusto, una porción pequeña de “outliers” (5 %) es suficiente para elevar el error de pose y geometría más allá de un valor útil. En contraposición, la columna de la derecha de la figura 5.16 muestra cómo la función de coste robusta mantiene el error aproximadamente constante en función de los outliers. En dicha columna se puede observar cómo el incremento en el número de cámaras disminuye el error de pose y geometría.

- Evaluación del error de inicialización en función del porcentaje de trayectoria utilizada como inicialización. El porcentaje de trayectoria utilizada varía en el rango siguiente:

$$\rho_t = (0,2 \quad 0,4 \quad 0,6 \quad 0,8 \quad 1)$$

. El resultado se muestra en las figuras 5.17.a, 5.17.b y 5.17.c.

En la figura 5.17 se muestra claramente cómo, mediante una trayectoria que consiste en al menos el 40 % de la circunferencia mostrada en la figura 5.15, se obtiene un valor de error que se mantiene constante, indicando la convergencia del algoritmo.

5.4. Conclusiones

En este capítulo se ha propuesto un sistema de inicialización del vector de estados que resuelve el problema de la obtención de la geometría y la pose del robot, necesarias para comenzar un proceso online de localización. Se plantea un algoritmo de reconstrucción diseñado para su funcionamiento mediante una o varias cámaras. Este requisito, se traduce en un método diferenciado cuando se tiene una o varias vistas del robot. En ambas alternativas se plantea que el robot realice una trayectoria de inicialización en la que se muestra al conjunto de cámaras el robot en diferentes poses.

En el caso de una sola cámara, el problema de la ambigüedad proyectiva se salva utilizando la odometría del robot mediante un criterio de máxima verosimilitud. Se ha demostrado que dicho criterio es capaz de mejorar la precisión obtenida en el caso usual de que la estimación de pose, dada por la odometría, presente ruido acumulativo. Se han propuesto varias alternativas de verosimilitud que tienen diferente influencia en el error cometido en la reconstrucción. Cada una de dichas alternativas da lugar a una función de coste que es minimizada con una técnica de “Bundle Adjustment”, la cual implica un algoritmo iterativo de tipo Gauss-Newton. Se han planteado soluciones a cada aspecto del problema, incluyendo el tiempo de cómputo, tratamiento de los “outliers”, inicialización del algoritmo iterativo de optimización y las configuraciones degeneradas de movimiento del robot.

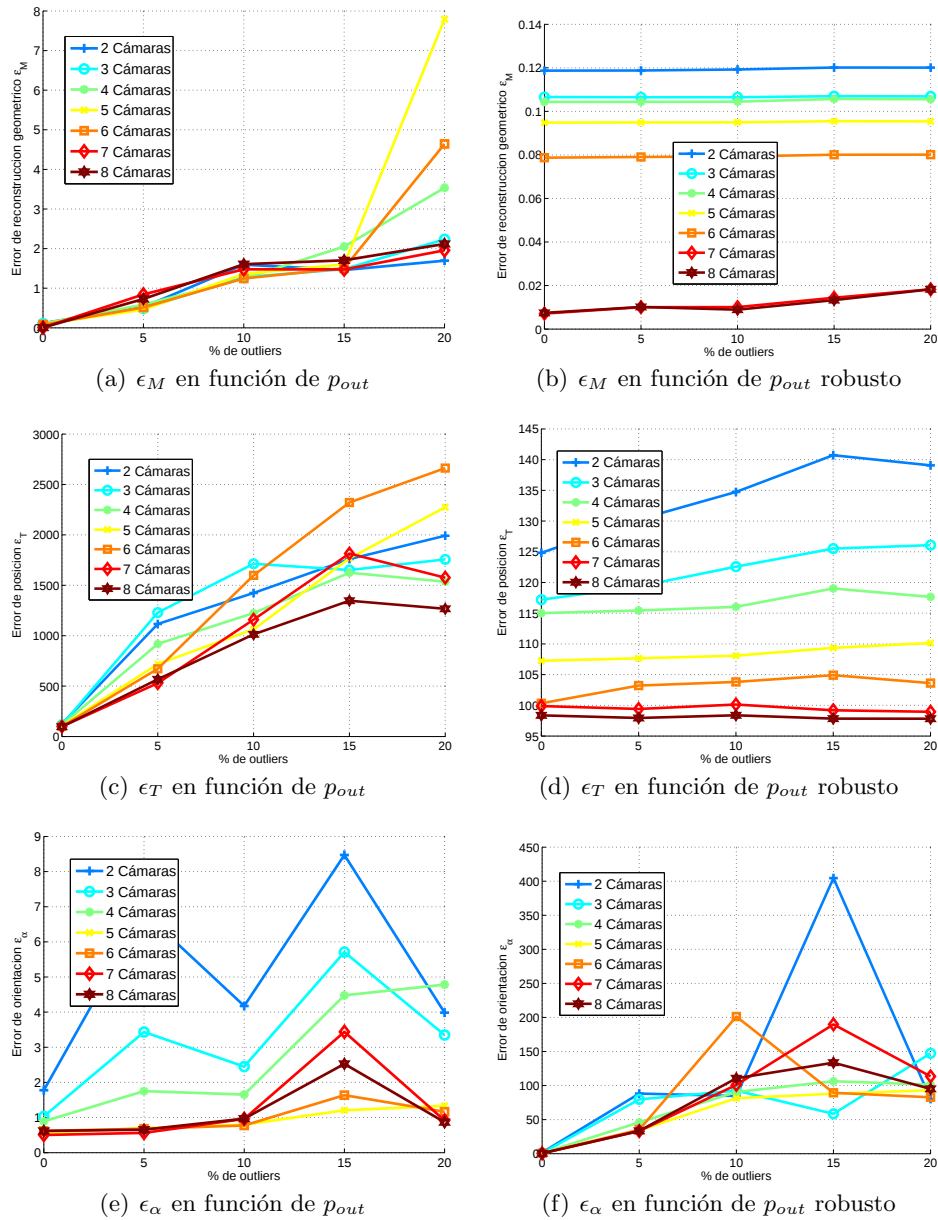


Figura 5.16: Experimento que muestra la variación del error de inicialización en función del porcentaje de “outliers” en la medida. En la columna de la izquierda y derecha se muestran los errores de reconstrucción y pose en el caso de no utilizar un método robusto y en el caso de utilizar la función robusta, respectivamente

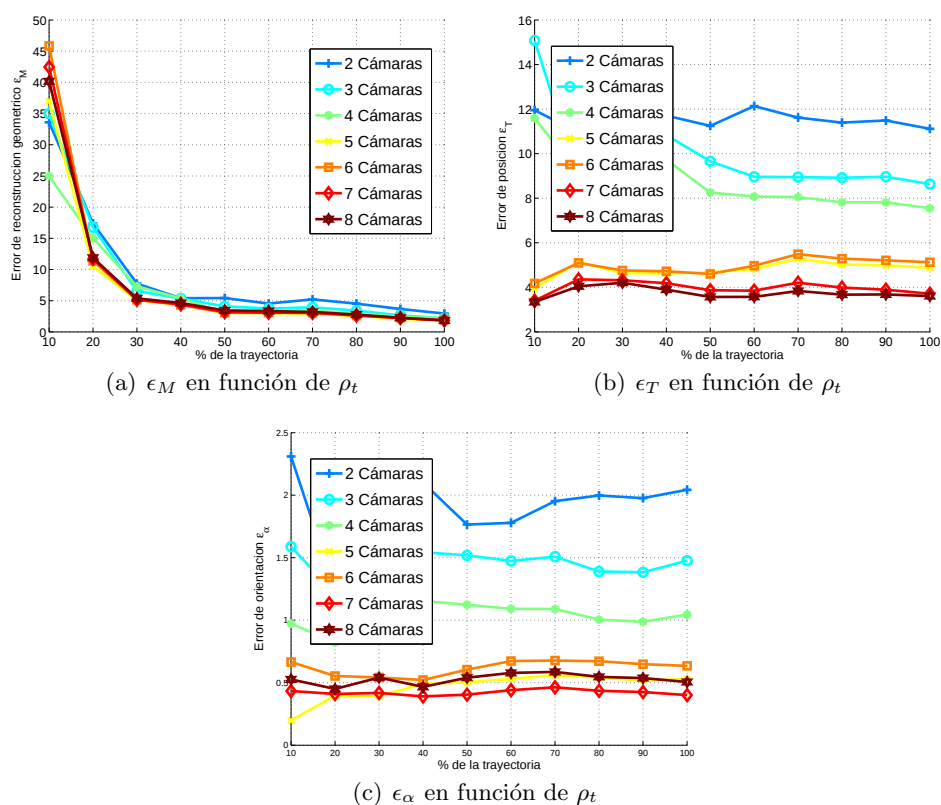


Figura 5.17: Experimento que muestra la variación del error de inicialización en función del porcentaje de trayectoria de inicialización utilizado.

El algoritmo de múltiples cámaras propone una alternativa a la inicialización, en la que no es necesaria la información proveniente de las correspondencias de un mismo punto en las diferentes cámaras. De este modo, se permitiría su uso en configuraciones de cámaras dónde el robot presenta vistas prácticamente complementarias. Al igual que en el caso de una única cámara, el algoritmo propuesto aborda cuestiones derivadas de la propuesta, como versiones robustas del algoritmo, inicialización, etc.

Las principales aportaciones del capítulo, referentes a la inicialización mediante una cámara se enumeran a continuación:

- Se ha propuesto un sistema de optimización, basado en una técnica de “Bundle Adjustment”, el cual se deriva de un criterio de máxima verosimilitud. Dicho criterio permite obtener una estimación del vector de estados en la que el impacto del error acumulativo de la odometría se ve mitigado considerablemente.
- Dentro del criterio de máxima verosimilitud, se han propuesto diversas alternativas, relacionadas con la estructura de la matriz Σ_L , con las que se obtiene una relación de compromiso entre precisión en la inicialización y el tiempo de proceso de cada propuesta.
- Mediante una función de coste robusta, se ha propuesto una versión del algoritmo inmune a posibles “outliers” en la medida.
- Se ha desarrollado una técnica no iterativa que permite resolver el problema de la inicialización del vector de estados previa a la optimización no lineal.

Las principales aportaciones relacionadas con la inicialización mediante múltiples cámaras se enumeran a continuación:

- El método presentado mediante de inicialización mediante múltiples cámaras permite obtener la pose y geometría del robot sin necesidad de recurrir a correspondencias entre múltiples vistas, lo cual es indispensable en situaciones con cámaras alejadas o incluso no solapadas.
- Se han propuesto una versión robusta del algoritmo que permite la presencia de “outliers” en la medida.
- Se han presentado dos métodos de inicialización previa a la optimización. Uno de ellos se basa en la utilización de la odometría, siendo equivalente al presentado mediante una sola cámara. El otro método, más genérico, permite obtener una primera aproximación del vector de estados sin recurrir a la odometría.

Capítulo 6

Obtención de la pose del robot a partir de múltiples cámaras

6.1. Introducción

El cálculo de la pose de un objeto a partir de correspondencias entre un modelo tridimensional del mismo y su proyección en una cámara de perspectiva se conoce como el problema PnP (“Perspective n Point problem”), el cual ha sido objeto de amplio estudio en la literatura. Sus aplicaciones se extienden a áreas relevantes como la fotogrametría, reconstrucción tridimensional, robótica móvil, etc.

Existen básicamente dos tipos de propuestas que ofrecen una solución al problema PnP. La primera consiste en algoritmos iterativos [Liu et al., 1990] [Phong et al., 1995], que obtienen la solución de la pose mediante la optimización de un criterio métrico (p. ej. minimizar la distancia entre las medidas y aquellas generadas artificialmente mediante la pose y el modelo). Suelen presentar una precisión muy alta siempre y cuando su convergencia se garantice. Para ello, en general requieren partir de un valor de la pose próximo al exacto, de tal manera que se minimice la probabilidad de caer en un extremo local. Su tiempo de cómputo es elevado cuando el número de correspondencias entre el modelo y la imagen es grande.

El segundo enfoque del problema, el cual es el objetivo en el presente capítulo, se centra con atención en algoritmos no iterativos y exactos, los cuales no requieran inicialización y no presenten problemas de convergencia. Las propuestas no iterativas del problema PnP han proliferado en la literatura [Nister and Stewenius, 2007], requiriendo en numerosas ocasiones dar solución a complejos sistemas de polinomios mediante el uso de conceptos de matemática simbólica (p.ej. resultantes de polinomios) [Quan and Lan, 1999] [Stewenius et al., 2005]. Dichas técnicas son especialmente adecuadas para resolver el problema PnP en sus casos límite, usando tan solo 3 o 4 correspondencias en la imagen pertenecientes al modelo. La solución de los casos límite o de número mínimo de correspondencias juega un papel importante en su aplicación a algoritmos robustos, que discriminan los “outliers” de un conjunto de medidas.

Muchos de los algoritmos no iterativos, diseñados para casos límite, no son adecuados para solucionar el problema cuando el número de correspondencias es grande, dada su complejidad y el hecho de que no se benefician del aumento en el número de puntos para obtener una medida de la pose de menor error.

Al contrario que en el caso de una sola cámara, en la literatura el problema PnP desde múltiples cámaras calibradas o mPnP (“Multiple Perspective n Point Problem”) no ha sido estudiado con tanta atención, aunque existen trabajos dedicados como el propuesto por [Chang and Chen, 2004] y recientemente se han ofrecido soluciones al problema para 3 y 4 puntos [Nister and Stewenius, 2007]. En la variante para múltiples cámaras, se plantea el problema de la estimación de la pose cuando las correspondencias pertenecen a varias cámaras, de las cuales se suponen conocidos sus parámetros de calibración. Las aplicaciones de esta situación a espacios inteligentes y en concreto al problema propuesto en esta tesis es fundamental.

Los métodos no iterativos, que solucionan el problema PnP en el caso de una sola cámara, son difícilmente aplicables al caso de múltiples cámaras calibradas, a excepción del método propuesto por [Nister and Stewenius, 2007] y aproximaciones de mínimos cuadrados lineales [Ess et al., 2007]. El principal inconveniente radica en el hecho de que

prácticamente todos los métodos exactos existentes solucionan la pose en el sistema de coordenadas de la cámara y suponen un solo centro de proyección. La inclusión directa de medidas de cámaras diferentes requiere utilizar un origen de coordenadas diferente y de manera general una cámara genérica con múltiples centros de proyección. Dada la dificultad de este hecho, en la literatura se propone a menudo restringir la solución al problema PnP en cada cámara por separado utilizando para ello los métodos existentes. Este tipo de soluciones desperdician información valiosa, como puede ser aquella en la que existen correspondencias entre cámaras.

En esta tesis se propone un método exacto y no iterativo para resolver el problema PnP generalizado a varias cámaras. Se presenta el problema genérico para N_Y correspondencias. Entre ellas, se distinguirán las correspondencias simples, en las que un punto del modelo tridimensional es visto por una cámara y las correspondencias múltiples, en las que un mismo punto es visto por más de dos cámaras. Siendo de especial interés para su utilización en algoritmos robustos, se comentará la particularización para el caso límite en el que $N_Y = 4$ correspondencias simples. Se mostrará a su vez la particularización del método propuesto en el caso de que la pose se encuentre restringida en un plano conocido.

6.2. Propuesta no iterativa de obtención de pose a partir de múltiples cámaras

La solución propuesta en esta tesis al problema mPnP amplía el método presentado en [Moreno-Noguer et al., 2007], en el cual se da solución al problema PnP con una sola cámara mediante un algoritmo no iterativo de complejidad $\mathcal{O}(N_Y)$, siendo N_Y el número de correspondencias en total detectadas en las cámaras. De entre todos los métodos exactos propuestos en la literatura, el reciente trabajo realizado en [Moreno-Noguer et al., 2007] permite utilizar un origen de referencia, desde el que se obtiene la pose del objeto, que no coincide necesariamente con el de la cámara. Este hecho permite obtener la pose mediante una o varias cámaras, con diferentes parámetros de calibración y además para un número genérico de correspondencias. El método propuesto se beneficia de manera natural de la inclusión de múltiples correspondencias para obtener menor error en la pose, ya que se basa en una solución de mínimos cuadrados.

En las siguientes secciones que componen este capítulo se comentará en primer lugar la parametrización utilizada y el cambio a coordenadas baricéntricas del modelo tridimensional, necesario en el desarrollo del algoritmo. En segundo lugar, el problema del computo de la pose se trasladará a dar solución a un sistema lineal, que, dependiendo de la circunstancia, será sobredeterminado o dará lugar a múltiples soluciones, las cuales serán reducidas a una única añadiendo restricciones existentes en el planteamiento. El método propuesto, aplicado a movimientos en un plano conocido, será expuesto a continuación, cerrando el capítulo con un análisis del error cometido, y una expresión analítica para obtener una caracterización estadística del mismo.

6.3. Parametrización en coordenadas baricéntricas

Se parte de un conjunto M formado por N puntos que componen el modelo tridimensional del robot o un subconjunto del mismo:

$$M = \{M^i \quad i = 1, \dots, N\} \quad (6.1)$$

Se definen 4 puntos de control, $C^1, \dots, C^4$, que forman un tetraedro, en cuyo interior deberán estar contenidos los puntos del modelo $M^i \quad i = 1, \dots, N$ (ver figura 6.1):

$$C^j = \begin{pmatrix} x_c^j \\ y_c^j \\ z_c^j \end{pmatrix} \quad j = 1, \dots, 4. \quad (6.2)$$

donde las coordenadas tridimensionales de los vértices del tetraedro se expresan mediante los símbolos x_c^j, y_c^j, z_c^j (la coordenada y_c^j no se deben confundir con los elementos del vector de medidas $y_k^{i,n}$, los cuales no utilizan el subíndice c).

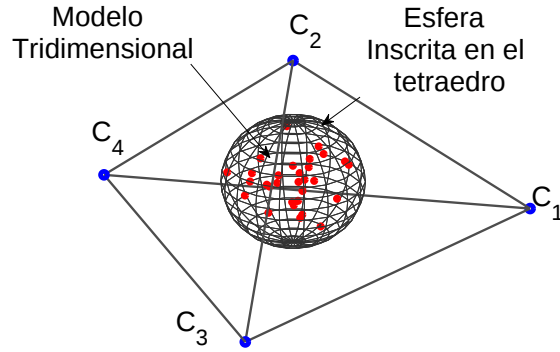


Figura 6.1: Tetraedro que inscribe a una esfera en la que estan contenidos todos los puntos del modelo

Cualquiera de los puntos en el modelo M^i se puede describir mediante la combinación lineal de los vértices del tetraedro:

$$M^i = \sum_{j=1}^4 \alpha_j^i C^j \quad \sum_{j=1}^4 \alpha_j^i = 1, \quad (6.3)$$

siendo $\alpha_j^i \quad j = 1, \dots, 4$ las coordenadas baricéntricas de los puntos del modelo, las cuales son fácilmente obtenidas dados los vértices del tetraedro. En la figura 6.2 se muestra

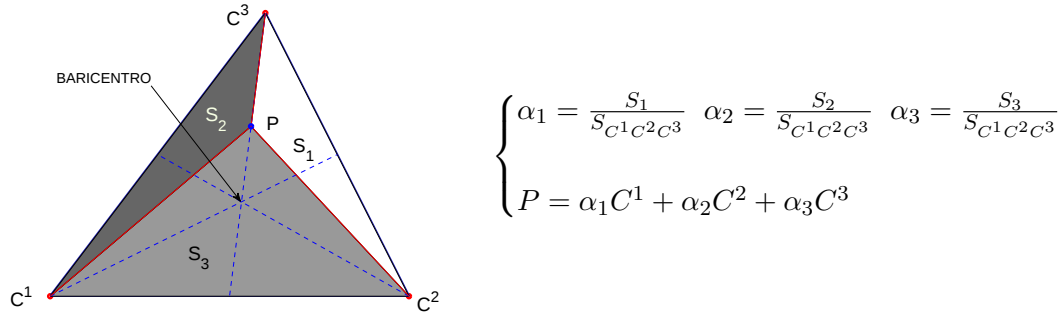
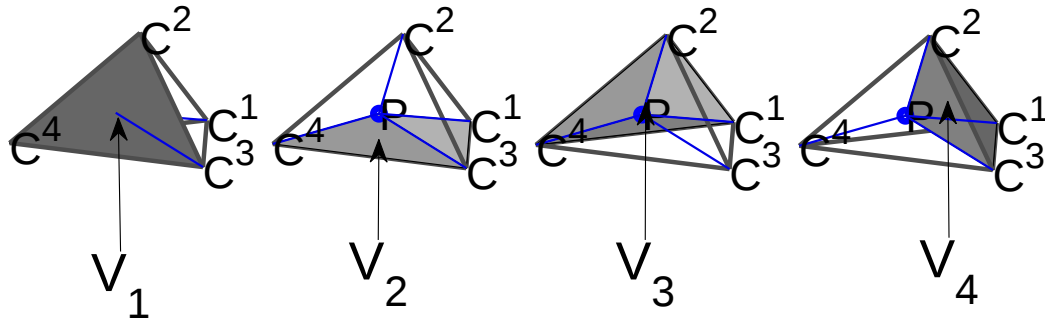


Figura 6.2: Coordenadas baricéntricas del punto P en el interior del triángulo definido por los vértices C^1, C^2, C^3 . Las coordenadas baricéntricas coincide con el valor de las áreas de los triángulos PC^3C^2 , PC^1C^3 y PC^1C^2 , normalizadas por el área total del triángulo $C^1C^2C^3$.



$$\begin{cases} \alpha_1 = \frac{V_1}{V_{C^1C^2C^3C^4}} & \alpha_2 = \frac{V_2}{V_{C^1C^2C^3C^4}} & \alpha_3 = \frac{V_3}{V_{C^1C^2C^3C^4}} & \alpha_4 = \frac{V_4}{V_{C^1C^2C^3C^4}} \\ P = \alpha_1 C^1 + \alpha_2 C^2 + \alpha_3 C^3 + \alpha_4 C^4 \end{cases}$$

Figura 6.3: Coordenadas baricéntricas del punto P en el interior del tetraedro definido por los vértices C^1, C^2, C^3, C^4 , y cuyo volumen se representa mediante $V_{C^1C^2C^3C^4}$. Las coordenadas baricéntricas en este caso se obtienen mediante los volúmenes $V_1, \dots, V_4$ encerrados por los 4 posibles tetrahedros, formados por el punto P y los 4 vértices.

gráficamente el cálculo de las coordenadas baricéntricas en el plano y en la figura 6.3, se muestra el equivalente en el espacio tridimensional.

En general cualquier posición de los vértices del tetraedro son válidas para representar un conjunto genérico de puntos en el espacio, siempre que se asegure que los puntos están contenidos en el tetraedro. Sin embargo, con el objetivo de mejorar la estabilidad del algoritmo, en [Moreno-Noguer et al., 2007] se propone un método que dispone los vértices del tetraedro en función de las direcciones principales en el espacio de los puntos del modelo.

Se define como M_k^i $i = 1, \dots, N$ a los puntos M^i transformados mediante una transformación afín, compuesta por una rotación R_k y una traslación T_k y definida por la pose.

Utilizando las coordenadas baricéntricas α_j^i se comprueba la expresión siguiente:

$$M_k^i = R_k M^i + T_k = \sum_{j=1}^4 \alpha_j^i (R_k C^j + T_k) = \sum_{j=1}^4 \alpha_j^i C_k^j, \quad (6.4)$$

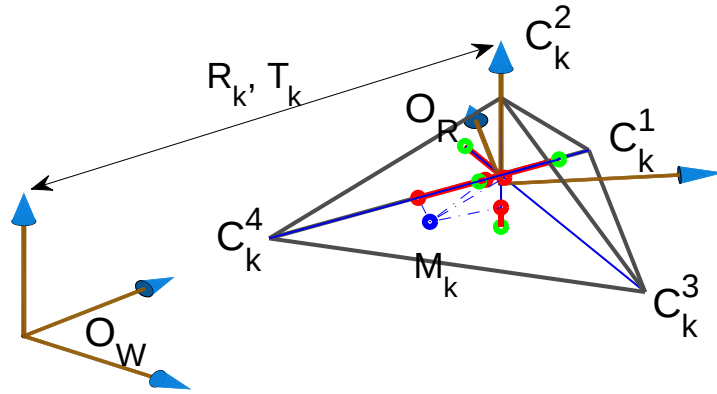


Figura 6.4: Las relaciones geométricas entre los vértices del tetraedro permanecen inalteradas ante una rotación y traslación, por lo que las coordenadas baricéntricas del punto M no varían.

donde se comprueba que el conjunto de puntos del modelo con respecto al origen de coordenadas del mundo $M_k^i, i = 1, \dots, N$ se definen mediante las mismas coordenadas baricéntricas calculadas a partir del conjunto $M^i, i = 1, \dots, N$, y utilizando los vértices del tetraedro transformados por la pose y denotados como C_k^j $j = 1, \dots, 4$.

El proceso que se propone a continuación tiene como objetivo el cálculo de R_k y T_k que

definen la pose del robot, a partir del conjunto M y las medidas en el plano imagen Y_k . Para ello, en lugar de plantear un sistema que relacione Y_k en términos de la pose, se propone resolver un sistema de ecuaciones en términos de coordenadas del tetraedro transformado por la pose C_k^j $j = 1, \dots, 4$. La relación entre Y_k y el tetraedro transformado se obtiene a partir de las coordenadas baricéntricas α_j^i $j = 1, \dots, 4$. Dado que las coordenadas baricéntricas permanecen inalteradas con los cambios de pose, éstas se calculan a partir del modelo disponible M y un tetraedro “base” (C^j $j = 1, \dots, 4$), en el que está contenido dicho modelo. De este modo, a partir de los vértices del tetraedro transformado C_k^j y mediante la expresión (6.4), se obtiene el conjunto de puntos del modelo M_k^i $i = 1, \dots, N$, transformados según la pose. A partir de la técnica propuesta por [Horn et al., 1988] se obtiene la matriz de rotación R_k y el vector de traslación T_k que cumplen que $M_k^i = R_k M^i + T_k \quad \forall i$. Finalmente, el cálculo de la pose X_k que da lugar a la rotación y traslación resulta sencillo.

6.4. Construcción del sistema lineal

Mediante el conjunto formado por N_c cámaras, se obtiene un vector de medidas Y_k , compuesto de N_Y correspondencias entre puntos del modelo tridimensional y medidas puntuales en el plano imagen de cualquiera de las cámaras. El número de correspondencias N_Y incluidas en Y_k , y el número de cámaras involucradas tendrán diverso impacto en el algoritmo propuesto. Es de gran interés el desarrollo de un algoritmo que sea versátil en el número de correspondencias y cámaras involucradas, ya que de este modo, será posible su uso en el problema concreto planteado, en el cual la información medida proviene de un número variable de cámaras.

Dada una medida cualquiera $y_k^{i,n}$, incluida en el vector Y_k , perteneciente a la cámara n , y correspondiente al punto M^i del modelo, se obtiene un sistema de ecuaciones lineales que la relaciona con la posición del tetraedro C_k^j $j = 1, \dots, 4$, transformado según la pose actual del robot X_k .

Siendo K_c^n, R_c^n, T_c^n , los parámetros de calibración de la cámara n , la medida $y_k^{i,n}$, incluida en el vector Y_k será:

$$\begin{pmatrix} y_k^{i,n} \\ 1 \end{pmatrix} = \lambda_k^{i,n} K_c^n \sum_{j=1}^4 \alpha_j^i (R_c^n C_k^j + T_c^n) \quad i = 1, \dots, N \quad n = 1, \dots, N_c \quad (6.5)$$

donde $\lambda_k^{i,n}$ es el factor de escala proyectivo. Desarrollando la ecuación (6.5) mediante la descripción de R_c^n como la concatenación de tres vectores fila F_1^n, F_2^n, F_3^n , el vector T_c^n en sus tres coordenadas T_x^n, T_y^n, T_z^n y considerando que $y_k^{i,n} = \begin{pmatrix} u_k^{i,n} & v_k^{i,n} \end{pmatrix}^T$ se obtiene la siguiente expresión:

$$\begin{pmatrix} u_k^{i,n} \\ v_k^{i,n} \\ 1 \end{pmatrix} = \lambda_k^{i,n} \begin{pmatrix} f_u^n & 0 & u_0^n \\ 0 & f_v^n & v_0^n \\ 0 & 0 & 1 \end{pmatrix} \sum_{j=1}^4 \alpha_j^i \begin{pmatrix} F_1^n \cdot C_k^j + T_x^n \\ F_2^n \cdot C_k^j + T_y^n \\ F_3^n \cdot C_k^j + T_z^n \end{pmatrix} \quad i = 1, \dots, N \quad n = 1, \dots, N_c \quad (6.6)$$

Las incógnitas del sistema son únicamente el conjunto de cuatro puntos del tetraedro C_k^j , que forman un total de 12 incógnitas y el valor de los N_Y factores de escala proyectivos $\lambda_k^{i,n}$, los cuales dependen de manera no lineal con la solución buscada. El valor de los factores de escala se puede eliminar de las incógnitas notando que la tercera fila de la expresión 6.6 resulta:

$$1 = \lambda_k^{i,n} \sum_{j=1}^4 \alpha_j^i (F_3^n \cdot C_k^j + T_z^n) \quad (6.7)$$

Sustituyendo la ecuación (6.7) en las dos primeras filas de (6.6) se obtiene el siguiente sistema de ecuaciones dependiente linealmente de los vértices del tetraedro:

$$\begin{aligned} \sum_{j=1}^4 \left[\alpha_j^i f_u^n (F_1^n \cdot C_k^j + T_x^n) + \alpha_j^i (u_0^n - u_k^{i,n}) (F_3^n \cdot C_k^j + T_z^n) \right] &= 0 \\ \sum_{j=1}^4 \left[\alpha_j^i f_v^n (F_2^n \cdot C_k^j + T_y^n) + \alpha_j^i (v_0^n - v_k^{i,n}) (F_3^n \cdot C_k^j + T_z^n) \right] &= 0 \end{aligned} \quad (6.8)$$

Denotando como vector de incógnitas $\Phi = \left(C_k^{1T} \ C_k^{2T} \ C_k^{3T} \ C_k^{4T} \right)^T$, y a partir de las expresiones expuestas en (6.8), se obtiene el correspondiente sistema lineal en Φ :

$$A\Phi = B \quad A \in \mathbb{R}^{2N_Y \times 12} \quad B \in \mathbb{R}^{2N_Y} \quad (6.9)$$

Se debe notar que para cada medida en el plano imagen y para cada cámara, el sistema obtiene dos ecuaciones linealmente independientes en función del vector Φ que contiene los cuatro vértices del tetraedro.

Utilizando la expresión (6.9) como base y en función de la situación concreta, el sistema propuesto será sobredeterminado, y por tanto se resolverá directamente mediante mínimos cuadrados lineales, o contendrá infinitas soluciones, en cuyo caso se añadirá información extra para resolver la ambigüedad.

El sistema propuesto en 6.9 tendrá una solución, ya sea sobredeterminado o necesitando restricciones extra, siempre que los puntos del modelo tridimensional cumplan la condición de no ser ni coplanares ni colineales.

Si se cumplen dichas condiciones en el modelo, el número de autovalores nulos del sistema dependerá de la configuración y tipo de correspondencias. Se distinguirán dos tipos de correspondencias en el vector de medidas Y_k :

1. Correspondencias simples: aquellos puntos del modelo que son vistos por una sola cámara.
2. Correspondencias dobles o múltiples: aquellos puntos del modelo que son vistos por dos o mas cámaras.

Se presentan a continuación dos resultados que serán importantes para clasificar los valores singulares nulos del sistema (6.9).

- **Resultado1 :** Si todas las correspondencias son simples existe al menos un autovalor nulo en la matriz $A^T A$:

Demostración. Sea un vector Y_k compuesto por medidas correspondientes a una sola cámara ($N_c = 1$), existe un conjunto de parámetros de calibración único P^1 , que da lugar a los parámetros extrínsecos T_c^1 y R_c^1 e intrínsecos K_c^1 .

Si $C_k^1, \dots, C_k^4$ es una solución, aglutinada en el vector columna Φ , al sistema representado por las ecuaciones (6.9), entonces la siguiente familia uniparamétrica $Cp_k^1(\nu), \dots, Cp_k^4(\nu)$ también es solución a dicho sistema:

$$Cp_k^i(\nu) = (\nu + 1)C_k^i + \nu(R_c^1)^T T_c^1 \quad (6.10)$$

Basta sustituir los puntos Cp_k^i en (6.8) para obtener el siguiente sistema:

$$A\Phi_p = (\nu + 1)A\Phi - \nu B = B \quad \rightarrow \quad (\nu + 1)A\Phi = (\nu + 1)B, \quad (6.11)$$

donde el vector Φ_p contiene los puntos Cp_k^i :

$$\Phi_p = \begin{pmatrix} Cp_k^1 \\ \vdots \\ Cp_k^4 \end{pmatrix}. \quad (6.12)$$

Dado que Φ es solución del sistema, Φ_p también lo es y el sistema tiene al menos un autovalor nulo. $\square$

- **Resultado2:** una correspondencia múltiple solo aporta 3 ecuaciones linealmente independientes al sistema:

Demostración. Sea un conjunto de medidas $y_k^{1,i}, \dots, y_k^{n,i}$, contenidas en el vector Y_k , que representan una correspondencia múltiple (punto M^i en n cámaras).

Cada una de las medidas aporta dos ecuaciones según la expresión 6.8, dando lugar

al siguiente conjunto:

$$\begin{aligned} \sum_{j=1}^4 \left[\alpha_j^i f_u^1 (F_1^1 \cdot C_k^j + T_x^1) + \alpha_j^i (u_0^1 - u_k^{i,1}) (F_1^1 \cdot C_k^j + T_z^1) \right] &= 0 \\ \sum_{j=1}^4 \left[\alpha_j^i f_v^1 (F_2^1 \cdot C_k^j + T_y^1) + \alpha_j^i (v_0^1 - v_k^{i,1}) (F_2^1 \cdot C_k^j + T_z^1) \right] &= 0 \end{aligned} \quad (6.13)$$

$$\vdots \quad \vdots \quad \vdots \quad (6.14)$$

$$\begin{aligned} \sum_{j=1}^4 \left[\alpha_j^i f_u^n (F_1^n \cdot C_k^j + T_x^n) + \alpha_j^i (u_0^n - u_k^{i,n}) (F_1^n \cdot C_k^j + T_z^n) \right] &= 0 \\ \sum_{j=1}^4 \left[\alpha_j^i f_v^n (F_2^n \cdot C_k^j + T_y^n) + \alpha_j^i (v_0^n - v_k^{i,n}) (F_2^n \cdot C_k^j + T_z^n) \right] &= 0 \end{aligned} \quad (6.15)$$

Sabiendo que $M_k^i = \sum_j \alpha_j^i C_k^j$ y que $\sum_j \alpha_j^i = 1$, entonces el sistema se simplifica en la siguiente expresión:

$$\begin{aligned} (f_u^1 F_1^1 + (u_0^1 - u_k^{i,1}) F_3^1) M_k^i &= -f_u^1 T_x^1 - (u_0^1 - u_k^{i,1}) T_z^1 \\ (f_v^1 F_2^1 + (v_0^1 - v_k^{i,1}) F_3^1) M_k^i &= -f_v^1 T_y^1 - (v_0^1 - v_k^{i,1}) T_z^1 \end{aligned} \quad (6.16)$$

$$\vdots \quad \vdots \quad \vdots \quad (6.17)$$

$$\begin{aligned} (f_u^n F_1^n + (u_0^n - u_k^{i,n}) F_3^n) M_k^i &= -f_u^n T_x^n - (u_0^n - u_k^{i,n}) T_z^n \\ (f_v^n F_2^n + (v_0^n - v_k^{i,n}) F_3^n) M_k^i &= -f_v^n T_y^n - (v_0^n - v_k^{i,n}) T_z^n \end{aligned} \quad (6.18)$$

,

donde el sistema se simplifica matricialmente como $A_M M_k^i = B_M$. Notando que:

$$M_k^i = M_{\alpha^i} \Phi \quad M_{\alpha^i} = \begin{pmatrix} \alpha_1^i, \dots, \alpha_4^i & \bar{0}_{1 \times 4} & \bar{0}_{1 \times 4} \\ \bar{0}_{1 \times 4} & \alpha_1^i, \dots, \alpha_4^i & \bar{0}_{1 \times 4} \\ \bar{0}_{1 \times 4} & \bar{0}_{1 \times 4} & \alpha_1^i, \dots, \alpha_4^i \end{pmatrix}, \quad (6.19)$$

entonces el sistema será:

$$A_M M_{\alpha^i} \Phi = B_M \quad A_M \in \mathbb{R}^{2n \times 3}, M_{\alpha^i} \in \mathbb{R}^{3 \times 12} \quad (6.20)$$

Puesto que el rango máximo de A_M es 3 y el rango máximo de M_{α^i} es 3, entonces la matriz del sistema $A = A_M M_{\alpha^i}$ también tiene un máximo de tres ecuaciones linealmente independientes, sin importar cuantas cámaras observan el mismo punto. $\square$

6.4.1. Sistema sobredeterminado o perfectamente determinado

En el caso de que el sistema expuesto en (6.9) sea sobredeterminado, la solución para los cuatro vértices del tetraedro se obtiene mediante la pseudo inversa de la matriz A , que resulta de la solución de mínimo error cuadrático del sistema:

$$\Phi = (A^T A)^{-1} A^T B \quad (6.21)$$

La solución Φ contiene coordenadas espaciales del tetraedro modificado por la pose y por tanto será exacta en el caso que el vector de medidas no contenga ruido. Obtenido el vector Φ , se utilizan las coordenadas baricéntricas resultantes α_j^i , $j = 1, \dots, 4$ para regenerar el conjunto de puntos del modelo tridimensional $M_k^1, \dots, M_k^N$, con respecto a la pose actual del robot:

$$M_k^i = \sum_{j=1}^4 \alpha_j^i C_k^j \quad i = 1, \dots, N \quad (6.22)$$

La pose actual del robot X_k resulta de la obtención de la rotación y translación existente entre el conjunto de puntos del modelo $M^1, \dots, M^N$ y aquellos calculados en (6.22) y referenciados con respecto a O_W :

$$M_k^i = R_k M^i + T_k \quad \forall i \quad (6.23)$$

En la literatura existen métodos exactos y sencillos para la obtención de las matrices R_k y T_k a partir de los pares $M_k^i \rightarrow M^i$. En esta tesis se propone usar la propuesta desarrollada en [Horn et al., 1988], la cual hace uso de cuaterniones de rotación para encontrar una solución directa y analítica al problema. Una vez obtenida la transformación afín R_k, T_k , se obtienen sin dificultad los elementos que componen el vector X_k .

El sistema lineal resultante en (6.9) está completamente determinado si el sistema tiene al menos 12 filas de la matriz A linealmente independientes. Acudiendo a los resultados anteriormente expuestos, si denominamos N_Y^s al número de correspondencias simples y N_Y^m al número de correspondencias múltiples, se necesita cumplir las condiciones siguientes:

$$\begin{cases} 2N_Y^s + 3N_Y^m \geq 12 \\ N_c > 1 \end{cases} \quad (6.24)$$

donde se entiende que por cada correspondencia múltiple se cuentan todos los puntos del vector Y_k que son imagen de un mismo punto del modelo del robot.

Ejemplo: se tiene un conjunto de $N_Y = N_Y^s \geq 6$ correspondencias simples, donde al menos una de ellas corresponde a una cámara diferente del resto.

6.4.2. Sistema con un único valor singular nulo

Se plantea en este punto la existencia de un único valor singular nulo en la matriz $A^T A$. En ese caso, el sistema admite una familia uniparamétrica de soluciones que se expresa como sigue:

$$\Phi = \Phi_0 + \lambda v \quad \lambda \in \mathbb{R} \quad (6.25)$$

La obtención de Φ_0 es directa a partir de la descomposición en valores singulares de $A^T A$ y el término independiente B . El vector v corresponde con el autovector de $A^T A$ asociado al único autovalor nulo presente.

La expresión (6.25) da lugar al siguiente conjunto de puntos C_k :

$$C_k^j = C_{0k}^j + \lambda C_{vk}^j \quad j = 1, \dots, 4 \quad (6.26)$$

Obviando configuraciones degeneradas de puntos del modelo o vértices del tetraedro, el sistema tiene en realidad una única solución asociada a un valor de λ concreto. Para encontrar dicho valor es necesario aplicar restricciones existentes en el problema y que no han sido impuestas en la construcción del sistema de ecuaciones.

Los puntos del tetraedro de referencia C^i , $i = 1, \dots, 4$ se relacionan mediante una rotación R_k y una traslación T_k , con aquellos contenidos en Φ y denominados C_k^i , $i = 1, \dots, 4$. Puesto que dicha transformación afín se compone de rotación y traslación, las distancias entre los puntos del tetraedro de referencia C se conservan en C_k .

Este hecho se traduce en la siguientes restricciones geométricas:

$$\|C^j - C^i\|^2 = \|C_k^j - C_k^i\|^2 \quad i \neq j, \quad (6.27)$$

siendo un total de 6 restricciones para 4 puntos del tetraedro.

Aplicando la restricción expresada en (6.27) a la solución obtenida en (6.25), se obtiene la siguiente relación cuadrática en el parámetro λ :

$$a_0 + a_1 \lambda + a_2 \lambda^2 = 0, \quad (6.28)$$

donde los coeficientes a_0 , a_1 y a_2 resultan de la siguiente expresión:

$$a_0 = \|C_{0k}^j - C_{0k}^i\|^2 - \|C^j - C^i\|^2 \quad (6.29)$$

$$a_1 = 2(C_{0k}^j - C_{0k}^i)^T (C_{vk}^j - C_{vk}^i) \quad (6.30)$$

$$a_2 = \|C_{vk}^j - C_{vk}^i\|^2 \quad (6.31)$$

Repitiendo el proceso para todas las restricciones de distancia, se obtiene el conjunto de 6 relaciones cuadráticas para el parámetro λ , siendo éste por tanto la solución de un sistema sobredeterminado de polinomios:

$$\begin{aligned} a_0^1 + a_1^1 \lambda + a_2^1 \lambda^2 &= 0 \\ &\vdots \\ a_0^6 + a_1^6 \lambda + a_2^6 \lambda^2 &= 0 \end{aligned} \quad (6.32)$$

Al igual que en el trabajo propuesto por [Moreno-Noguer et al., 2007], se propone obtener la solución de mínimos cuadrados del sistema (6.32) mediante la técnica de la

relinealización. Dicha técnica convierte el sistema de polinomios en un sistema lineal el cual es resuelto de manera convencional.

Se define el vector $\Psi = (\lambda, \lambda^2)^T$, y el sistema lineal equivalente al expuesto en (6.32) en Ψ como:

$$A_\Psi \Psi - B_\Psi = 0 \quad A_\Psi = \begin{pmatrix} a_1^1 & a_2^1 \\ \vdots & \vdots \\ a_1^6 & a_2^6 \end{pmatrix} \quad B_\Psi = \begin{pmatrix} -a_0^1 \\ \vdots \\ -a_0^6 \end{pmatrix} \quad (6.33)$$

El sistema expuesto en (6.33) se resuelve mediante la pseudoinversa de la matriz A_Ψ , obteniendo así el vector Ψ y con él, el valor de λ que establece una única solución para los puntos del tetraedro:

$$\Psi = (A_\Psi^T A_\Psi)^{-1} A_\Psi^T B \quad \lambda = \Psi(1) \quad (6.34)$$

En determinadas configuraciones la matriz A_Ψ puede ser de rango unidad y por tanto la solución de mínimos cuadrados no está definida. En dicho caso, se obtiene la solución del parámetro λ mediante la solución directa de las raíces de uno de los polinomios que forman el sistema a partir de sus coeficientes. De dicho polinomio existen dos soluciones posibles para el parámetro λ :

$$\lambda_1 = \frac{-a_1 + \sqrt{(a_1^2 - 4a_1a_0)}}{a_0} \quad \lambda_2 = \frac{-a_1 - \sqrt{(a_1^2 - 4a_1a_0)}}{a_0} \quad (6.35)$$

Uno de los parámetros λ posibles da lugar a un tetraedro que se encuentra detrás del plano imagen de la cámara y por tanto es fácilmente descartable.

Configuraciones que dan lugar a un autovalor nulo

Las situaciones en las que se encuentra un único autovalor nulo se resumen en las siguientes condiciones:

$$\begin{cases} 2N_Y^s + 3N_Y^m = 11 \\ N_c > 2 \end{cases} \quad o \quad \begin{cases} 2N_Y^s + 3N_Y^m > 11 \\ N_c = 1 \end{cases} \quad (6.36)$$

es decir, si el número de filas linealmente independiente es 11 o si se usa una única cámara. La expresión 6.36 se cumple en los siguientes ejemplos:

- Se tienen $N_Y^s > 5$ correspondencias simples, pertenecientes a la misma cámara. *Ejemplo: Una sola cámara $n = 1$, $N = 10$ puntos del modelo y $N_Y^s = 10$ correspondencias simples*

$$Y_k = (y_k^{1,1}, \dots, y_k^{1,10}).$$

- Se tienen $N_Y^s = 4$ y $N_Y^m = 1$.

Ejemplo: Dos cámaras $n \in \{1, 2\}$, $N = 5$ puntos del modelo y $N_Y = 6$ correspondencias

$$Y_k = (y_k^{1,1}, y_k^{1,2}, \boxed{y_k^{1,3}, y_k^{2,3}}, y_k^{2,4}, y_k^{2,5}).$$

Ejemplo: Cinco cámaras $n \in \{1, 2, 3, 4, 5, 6\}$, $N = 5$ puntos del modelo y $N_Y = 6$ correspondencias

$$Y_k = (y_k^{1,1}, y_k^{2,2}, \boxed{y_k^{3,3}, y_k^{4,3}}, y_k^{5,4}, y_k^{6,5}).$$

- Se tienen $N_Y^s = 1$ y $N_Y^m = 3$. *Ejemplo:* Tres cámaras $n \in \{1, 2, 3\}$, $N = 4$ puntos del modelo y $N_Y = 7$ correspondencias

$$Y_k = (y_k^{1,1}, y_k^{1,2}, y_k^{1,3}, y_k^{2,1}, y_k^{2,2}, y_k^{2,3}, y_k^{3,1}).$$

6.4.3. Sistema con dos valores singulares nulos

En este caso, A posee dos valores singulares nulos y por tanto existe la siguiente familia de soluciones biparamétrica:

$$\Phi = \Phi_0 + \lambda_1 v_1 + \lambda_2 v_2 \quad \lambda_1, \lambda_2 \in \mathbb{R}, \quad (6.37)$$

que a su vez da lugar al siguiente conjunto de puntos C_k :

$$C_k^j = C_{0k}^j + \lambda_1 C_{v_1k}^j + \lambda_2 C_{v_2k}^j \quad j = 1, \dots, 4 \quad (6.38)$$

Aplicando una restricción de distancia cualquiera de las seis disponibles (6.27), se obtiene la siguiente ecuación cuadrática:

$$a_{0,0} + a_{1,0}\lambda_1 + a_{0,1}\lambda_2 + a_{1,1}\lambda_1\lambda_2 + a_{2,0}\lambda_1^2 + a_{0,2}\lambda_2^2 = 0, \quad (6.39)$$

donde los coeficientes a_0 , a_1 y a_2 resultan de la siguiente expresión:

$$\begin{aligned} a_{0,0} &= \|C_{0k}^j - C_{0k}^i\|^2 - \|C^j - C^i\|^2 \\ a_{1,0} &= 2(C_{0k}^j - C_{0k}^i)^T (C_{v_1k}^j - C_{v_1k}^i) \\ a_{0,1} &= 2(C_{0k}^j - C_{0k}^i)^T (C_{v_2k}^j - C_{v_2k}^i) \\ a_{1,1} &= 2(C_{v_1k}^j - C_{v_1k}^i)^T (C_{v_2k}^j - C_{v_2k}^i) \\ a_{2,0} &= \|C_{v_1k}^j - C_{v_1k}^i\|^2 \\ a_{0,2} &= \|C_{v_2k}^j - C_{v_2k}^i\|^2 \end{aligned} \quad (6.40)$$

En este caso la relinealización consta de 5 parámetros en el vector $\Psi = (\lambda_1, \lambda_2, \lambda_1\lambda_2, \lambda_1^2, \lambda_2^2)^T$ y 6 restricciones que forman el sistema A_Ψ, B_Ψ :

$$A_{\Psi}\Psi - B_{\Psi} = 0 \quad A_{\Psi} = \begin{pmatrix} a_{1,0}^1 & a_{0,1}^1 & a_{1,1}^1 & a_{2,0}^1 & a_{0,2}^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{1,0}^6 & a_{0,1}^6 & a_{1,1}^6 & a_{2,0}^6 & a_{0,2}^6 \end{pmatrix} \quad B_{\Psi} = \begin{pmatrix} -a_{0,0}^1 \\ \vdots \\ -a_{0,0}^6 \end{pmatrix} \quad (6.41)$$

Si al menos existen 5 de las 6 restricciones linealmente independientes, el vector Ψ se obtiene de manera general mediante la expresión de la pseudoinversa de la matriz A_{Ψ} .

$$\Psi = (A_{\Psi}^T A_{\Psi})^{-1} A_{\Psi}^T B_{\Psi} \quad \lambda_1 = \Psi(1) \quad \lambda_2 = \Psi(2) \quad (6.42)$$

En el caso de que la matriz $A_{\Psi}^T A_{\Psi}$ sea singular, se añaden restricciones que se obvian en la relinealización y que tratan las relaciones no lineales existentes entre los propios elementos del vector Ψ (p. ej. $\Psi(1)\Psi(2) = \Psi(3)$). Dicha técnica se introduce en el siguiente punto, ya que con más de dos autovalores nulos, será siempre necesario su uso.

Configuraciones que dan lugar a dos autovalores nulos

Las situaciones en las que se tienen dos valores singulares nulos en la matriz M son las siguientes:

$$\{2N_Y^s + 3N_Y^m = 10 \quad (6.43)$$

Las situaciones en las que se cumple la condición (6.43) se muestran a continuación:

- Se tienen $N_Y = 5$ correspondencias simples, pertenecientes a una o varias cámaras.

Ejemplo: Una sola cámara $n = 1$, $N = 5$ puntos del modelo y $N_Y = 5$ correspondencias

$$Y_k = (y_k^{1,1}, \dots, y_k^{1,5}).$$

Ejemplo: Cuatro cámaras $n \in \{1, 2, 3, 4\}$, $N = 5$ puntos del modelo y $N_Y = 5$ correspondencias

$$Y_k = (y_k^{1,1}, y_k^{2,2}, y_k^{3,3}, y_k^{4,4}, y_k^{4,5}).$$

- Se tienen $N_Y^s = 2$ y $N_Y^m = 2$. *Ejemplo: Cuatro cámaras $n \in \{1, 2, 3, 4\}$, $N = 4$ puntos del modelo y $N_Y = 5$ correspondencias*

$$Y_k = (y_k^{1,1}, y_k^{1,2}, y_k^{2,3}, y_k^{3,3}, y_k^{4,4}, y_k^{5,4}).$$

6.4.4. Sistema con más de dos valores singulares

En este punto se tratan los casos en los que se tienen 3 o 4 valores singulares, producidos por situaciones en las que se cubre completamente la casuística de correspondencias en el vector de medidas, incluidas aquellas con un mínimo de 4 correspondencias simples del

modelo. Se plantea con más detalle a continuación la solución con 3 valores singulares nulos.

En este caso existe una familia de soluciones de tres parámetros:

$$\Phi = \Phi_0 + \lambda_1 v_1 + \lambda_2 v_2 + \lambda_3 v_3 \quad \lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}, \quad (6.44)$$

dando lugar al siguiente conjunto de puntos C_k^j $j = 1, \dots, 4$:

$$C_k^j = C_{0k}^j + \lambda_1 C_{v_1 k}^j + \lambda_2 C_{v_2 k}^j + \lambda_3 C_{v_3 k}^j \quad j = 1, \dots, 4 \quad (6.45)$$

Mediante las restricciones de distancia de los puntos del tetraedro se obtiene un sistema A_Ψ linealizado de 6 ecuaciones sobre un vector Ψ de 9 incógnitas:

$$\Psi = (\lambda_1 \quad \lambda_2 \quad \lambda_3 \quad \lambda_1 \lambda_2 \quad \lambda_1 \lambda_3 \quad \lambda_2 \lambda_3 \quad \lambda_1^2 \quad \lambda_2^2 \quad \lambda_3^2)^T \quad (6.46)$$

El sistema está infradeterminado y por tanto la matriz $A_\Psi^T A_\Psi$ tiene como mínimo 3 valores singulares nulos. Existe por tanto una familia, que supondremos en un caso general de 3 parámetros, que define la solución al vector Ψ :

$$\Psi = \Psi_0 + \delta_1 \Psi_{v_1} + \delta_2 \Psi_{v_2} + \delta_3 \Psi_{v_3} \quad (6.47)$$

La indeterminación de Ψ proviene del aumento de parámetros que requiere la relinealización del sistema de polinomios, y se puede resolver el valor exacto de λ_1 , λ_2 y λ_3 aplicando restricciones a los elementos que lo componen.

Para representar de una manera compacta dichas restricciones, se propone la siguiente descripción de los elementos que componen el vector Ψ :

$$\Psi = (\Psi_{1,0,0} \quad \Psi_{0,1,0} \quad \Psi_{0,0,1} \quad \Psi_{1,1,0} \quad \Psi_{1,0,1} \quad \Psi_{0,2,1} \quad \Psi_{2,0,0} \quad \Psi_{0,2,0} \quad \Psi_{0,0,2})^T, \quad (6.48)$$

donde cada elemento se indexa por tres coordenadas que indican el exponente correspondiente al parámetro λ . De este modo $\Psi_{1,0,1} = \lambda_1 \lambda_3$ ó $\Psi_{2,0,0} = \lambda_1^2$. Mediante dicha numeración, las restricciones corresponden a la siguiente expresión:

$$\Psi_{i_1, j_1, k_1} \Psi_{i_2, j_2, k_2} = \Psi_{i_3, j_3, k_3} \Psi_{i_4, j_4, k_4} \quad (6.49)$$

$$(6.50)$$

$$\begin{aligned} i_1 + i_2 &= i_3 + i_4 & i_1 + i_2, i_3 + i_4 &= 0, 1, 2 \\ j_1 + j_2 &= j_3 + j_4 & i_1 + i_2, i_3 + i_4 &= 0, 1, 2 \\ k_1 + k_2 &= k_3 + k_4 & i_1 + i_2, i_3 + i_4 &= 0, 1, 2, \end{aligned} \quad (6.51)$$

donde se supone que el elemento $\Psi_{0,0,0} = 1$ y se deben eliminar todos aquellos casos que se repiten bien debido a la propiedad conmutativa de la multiplicación (p. ej. $\Psi_{0,0,1} \Psi_{1,0,0} =$

$\Psi_{1,0,0}\Psi_{0,0,1}$) o bien aquellos debidos a un intercambio entre los valores de los coeficientes en ambos lados del signo de igualdad (p. ej. $\Psi_{0,0,1}\Psi_{1,0,0} = \Psi_{1,0,1} \rightarrow \Psi_{1,0,1} = \Psi_{0,0,1}\Psi_{1,0,0}$).

Cada restricción, resultado de la expresión (6.49), da lugar a una ecuación polinómica de segundo orden en las variables $\delta_1, \delta_2, \delta_3$.

Mediante una nueva relinealización se obtiene un sistema lineal en el vector de incógnitas Υ :

$$\Upsilon = (\delta_1, \delta_2, \delta_3, \delta_1\delta_2, \delta_1\delta_3, \delta_2\delta_3, \delta_1^2, \delta_2^2, \delta_3^2) \quad (6.52)$$

$$A_\Upsilon \Upsilon = B_\Upsilon \quad \Upsilon = (A_\Upsilon^T A_\Upsilon)^{-1} A_\Upsilon^T B_\Upsilon \quad (6.53)$$

Obtenido el vector Υ , se compone la solución para el vector Ψ utilizando la expresión (6.40) y de nuevo la solución para los puntos del tetraedro mediante la expresión (6.38).

En el caso de que se tengan 4 autovalores nulos en el sistema (6.9), el desarrollo es exacto al propuesto para 3, con la salvedad del aumento de variables requeridas en el vector Ψ , que asciende a 14 y en el número de autovalores nulos de A_Ψ , que asciende a 8. El homólogo de la expresión (6.49) corresponde con la siguiente expresión:

$$\begin{aligned} \Psi_{i_1, j_1, k_1, n_1} \Psi_{i_2, j_2, k_2, n_2} &= \Psi_{i_3, j_3, k_3, n_3} \Psi_{i_4, j_4, k_4, n_4} \\ i_1 + i_2 = i_3 + i_4 &\rightarrow i_1 + i_2 = 0, 1, 2 \quad i_3 + i_4 = 0, 1, 2 \\ j_1 + j_2 = j_3 + j_4 &\rightarrow j_1 + j_2 = 0, 1, 2 \quad j_3 + j_4 = 0, 1, 2 \\ k_1 + k_2 = k_3 + k_4 &\rightarrow k_1 + k_2 = 0, 1, 2 \quad k_3 + k_4 = 0, 1, 2 \\ n_1 + n_2 = n_3 + n_4 &\rightarrow n_1 + n_2 = 0, 1, 2 \quad n_3 + n_4 = 0, 1, 2 \end{aligned} \quad (6.54)$$

donde de manera equivalente se obtiene un sistema de polinomios que se resuelve mediante una nueva relinealización usando el vector Υ , cuyo tamaño asciende a 20.

En ambos casos, 3 o 4 autovalores nulos, existen determinadas configuraciones en las que el sistema A_Υ nuevamente posee al menos un autovalor nulo. En dichos casos, puesto que el vector Υ proviene de una relinealización de un sistema de polinomios, admite un conjunto de restricciones, como se hizo con el vector Ψ . Mediante la aplicación de nuevo del proceso al vector Υ , se obtiene un sistema determinado que admite solución.

Configuraciones que dan lugar tres autovalores nulos

Las situaciones que requieren resolver el problema de 3 autovalores aparecen bajo las circunstancias siguientes:

$$\begin{cases} 2N_Y^s + 3N_Y^m = 9 \\ N_c > 1 \end{cases} \quad (6.55)$$

- Se tiene $N_Y^s = 3$ y $N_Y^m = 1$.

Ejemplo: Dos cámaras $n \in \{1, 2\}$, $N = 4$ puntos del modelo y $N_Y = 5$ correspondencias

$$Y_k = (y_k^{1,1}, y_k^{1,2}, y_k^{1,3}, y_k^{2,3}, y_k^{2,4}).$$

Configuraciones que dan lugar a cuatro autovalores nulos

El problema que genera 4 autovalores corresponde al caso siguiente:

$$\begin{cases} 2N_Y^s + 3N_Y^m = 8 \end{cases}, \quad (6.56)$$

donde puesto que el número de puntos del modelo tridimensional debe ser $N > 3$, ya que de otro modo estaría contenido en un plano, la condición 6.56 implica que $N_Y^m = 0$ y por tanto se simplifica a exigir que $N^s = N_Y = 4$.

- Se tiene un vector de medidas de $N_Y = 4$ correspondencias simples en una o varias cámaras.

Ejemplo: Dos cámaras $n \in \{1, 2\}$, $N = 4$ puntos del modelo y $N_Y = 4$ correspondencias

$$Y_k = (y_k^{1,1}, y_k^{1,2}, y_k^{2,3}, y_k^{2,4}).$$

Ejemplo: Una cámara $n = 1$, $N = 4$ puntos del modelo y $N_Y = 4$ correspondencias

$$Y_k = (y_k^{1,1}, y_k^{1,2}, y_k^{1,3}, y_k^{1,4}).$$

La solución al problema de los cuatro puntos corresponde al problema límite y mediante el método propuesto es resuelto para una o varias cámaras.

6.5. Solución para movimientos restringidos a un plano

En aquellas configuraciones en las que el movimiento del objeto esta restringido en un plano conocido, como es el caso de un robot con ruedas en su movimiento por el plano del suelo, el algoritmo planteado se simplifica considerablemente y requiere de ciertas modificaciones que se presentan en esta sección.

Se supone que el plano, al cual está restringido el movimiento, coincide con el plano $x^3 = 0$ en el sistema de coordenadas del mundo O_W . En caso de que el plano no coincida con el utilizado en el eje x^3 , será suficiente con transformar el origen de coordenadas del mundo O_W de tal manera que coincidan.

6.5.1. Construcción del sistema lineal

Si el robot se mueve en el plano $x^3 = 0$, la pose estará definida por un ángulo de rotación α_k en el eje x^3 y la posición en el plano x^1, x^2 . Las matrices R_k y T_k de la expresión (6.23), las cuales representan el movimiento del robot, serán:

$$R_k = \begin{pmatrix} \cos(\alpha_k) & -\sin(\alpha_k) & 0 \\ \sin(\alpha_k) & \cos(\alpha_k) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad T_k = \begin{pmatrix} T_{x^1,k} \\ T_{x^2,k} \\ 0 \end{pmatrix}. \quad (6.57)$$

Como consecuencia, las coordenadas en el eje x^3 del tetraedro (transformado por la pose) que se busca $z_{c,k}^j, j = 1, \dots, 4$ permanecerán invariantes con respecto a las obtenidas en el sistema de coordenadas del robot $z_c^j, j = 1, \dots, 4$. De este modo, el sistema lineal expuesto en (6.8) posee en este caso como incógnitas tan solo las coordenadas en el plano de cada vértice del tetraedro:

$$\Phi_p = (x_{c,k}^1 \quad y_{c,k}^1 \quad x_{c,k}^2 \quad y_{c,k}^2 \quad x_{c,k}^3 \quad y_{c,k}^3 \quad x_{c,k}^4 \quad y_{c,k}^4)^T \quad (6.58)$$

Si se descompone la matriz A original mostrada en (6.9) en sus doce columnas:

$$A = (A_{C_1} \quad \dots \quad A_{C_{12}}), \quad (6.59)$$

entonces el sistema $A_p \Phi_p = B_p$ resultante de incluir como constantes las coordenadas $z_{c,k}^j$, en las ecuaciones (6.8) será:

$$A_p = (A_{C_1} \quad A_{C_2} \quad A_{C_4} \quad A_{C_5} \quad A_{C_7} \quad A_{C_8} \quad A_{C_{10}} \quad A_{C_{11}}) \quad (6.60)$$

$$B_p = B - z_c^1 A_{C_3} - z_c^2 A_{C_6} - z_c^3 A_{C_9} - z_c^4 A_{C_{12}} \quad (6.61)$$

6.5.2. Sistema sobredeterminado o perfectamente determinado

Si el rango de la matriz A_p es máximo, la solución a los vértices del tetraedro en el plano $x^3 = 0$, se obtiene, de forma general, mediante la pseudoinversa de la matriz A_p , de manera análoga a la expresión (6.21):

$$\Phi_p = (A_p^T A_p)^{-1} A_p^T B_p. \quad (6.62)$$

Los puntos del tetraedro, transformados según la pose actual se compondrán por tanto de la solución obtenida en Φ_p y las coordenadas en el eje z , que permanecen constantes:

$$C_k^j = (x_{c,k}^j \quad y_{c,k}^j \quad z_c^j) \quad j = 1, \dots, 4 \quad (6.63)$$

Mediante la expresión (6.22), se obtienen las coordenadas de los puntos del modelo tridimensional del robot con respecto a la pose actual $M_k^i, i = 1, \dots, N$. En este caso, tal y como se ha mostrado en (6.57), la rotación y traslación que define la pose, y que relacionan M_k^i con M^i , tiene una estructura simplificada. Su obtención se realiza con una versión modificada del algoritmo comentado en [Horn et al., 1988] y aplicado al caso de movimientos en un plano.

El sistema propuesto posee solución directa siempre que se dispongan de al menos 4 correspondencias simples entre el modelo y las medidas y que éstas no correspondan a puntos coplanares. La propuesta restringida en un plano permite por tanto determinar la pose, mediante una o varias cámaras, sin necesidad de obtener descomposiciones en valores singulares, como sí ocurría en el caso de movimiento no restringido.

Cuando el movimiento del objeto se realiza en un plano es posible en teoría obtener la pose unívocamente mediante un número inferior a 4 correspondencias. Sin embargo en dichos casos, que corresponden a modelos compuestos por 3 e incluso 2 puntos, éstos son necesariamente coplanares o colineales, lo cual no impone suficientes restricciones para calcular los vértices del tetraedro. El algoritmo propuesto para dichos casos particulares, requiere modificaciones esenciales, las cuales serán válidas para dar adicionalmente soluciones al problema en caso de coplanaridad y colinealidad.

6.5.3. Cómputo de pose para puntos coplanares

En este punto se describe el algoritmo de obtención de la pose de un objeto que se mueve en el plano $x^3 = 0$ y cuyo modelo tridimensional se compone de puntos coplanares. El plano Π , al cual pertenecen todos los puntos, se define por su vector director n^Π y su distancia al origen d^Π . De esta manera la siguiente relación se cumple:

$$n^\Pi M^i + d^\Pi = 0 \quad i = 1, \dots, N \quad (6.64)$$

Puesto que los puntos M^i se encuentran contenidos en un plano, se contruye un triángulo cualquiera C^1, C^2, C^3 , que reemplaza al tetraedro de la sección anterior, tal que el conjunto de puntos se encuentre siempre dentro del mismo.

El conjunto de puntos se puede describir mediante las coordenadas baricéntricas en dicho triángulo, como fue mostrado en la figura 6.2. Es decir, por cada M^i existe un vector de tres coordenadas baricéntricas $\alpha^i = (\alpha_1^i, \alpha_2^i, \alpha_3^i)$ tal que:

$$M^i = \alpha_1^i C^1 + \alpha_2^i C^2 + \alpha_3^i C^3 \quad \alpha_1^i + \alpha_2^i + \alpha_3^i = 1 \quad (6.65)$$

Al igual que se propuso con los vértices del tetraedro, en este caso ante un cambio en la pose del objeto, las coordenadas del triángulo $C_k^j \dots j = 1, 2, 3$ bastan para definir la pose del robot X_k y los puntos M_k^i transformados.

En este caso se construye un vector de nueve incógnitas $\Phi = ((C_k^1)^T, (C_k^2)^T, (C_k^3)^T)$, compuesto por las coordenadas de los tres puntos del tetraedro. Utilizando las mismas ecuaciones presentadas en (6.8), pero particularizadas para utilizar tres puntos de control en lugar de cuatro, que corresponderían al tetraedro, se construye sin dificultad un sistema lineal equivalente para el vector Φ :

$$A\Phi = B \quad A \in \mathbb{R}^{2N_Y \times 9} \quad B \in \mathbb{R}^{2N_Y} \quad (6.66)$$

Puesto que el robot se mueve en el plano $x^3 = 0$, se propone de nuevo transformar el sistema (6.66) en uno con seis incógnitas:

$$\Phi_p = (x_{c,k}^1, y_{c,k}^1, x_{c,k}^2, y_{c,k}^2, x_{c,k}^3, y_{c,k}^3)^T, \quad (6.67)$$

donde x_k^j, y_k^j representan las coordenadas en el plano $x^3 = 0$ de los puntos del triángulo. Del mismo modo, descomponiendo A en sus nueve columnas $A_{C_1}, \dots, A_{C_9}$, se obtienen las matrices A_p y B_p

$$A_p = (A_{C_1} \ A_{C_2} \ A_{C_4} \ A_{C_5} \ A_{C_7} \ A_{C_8}) \quad (6.68)$$

$$B_p = B - (A_{C_3}z_c^1 + A_{C_6}z_c^2 + A_{C_9}z_c^3) \quad (6.69)$$

El sistema A_p, B_p está determinado incluso con al menos 3 puntos del modelo (siempre y cuando no sean colineales), vistos por un número genérico de cámaras que aportan al menos 3 correspondencias simples de los mismos.

6.5.4. Cómputo de pose para puntos colineales

Si el conjunto de puntos forma una recta en el espacio tridimensional y el robot se mueve en el plano $x^3 = 0$, existe la posibilidad de obtener la pose. En este caso, puesto que los puntos son colineales, es suficiente encontrar dos puntos C^1, C^2 de la recta que forman, entre los cuales estén contenidos todos aquellos del modelo. Las coordenadas α^i en este caso se compondrán de dos valores que determinan unívocamente la posición de cualquier punto del modelo:

$$M^i = \alpha_1^i C^1 + \alpha_2^i C^2 \quad \alpha_1^i + \alpha_2^i = 1 \quad i = 1, \dots, N \quad (6.70)$$

De un modo totalmente análogo al caso anterior, se construye un sistema lineal que tiene como incógnitas las coordenadas tridimensionales de los dos puntos de control C_k^1, C_k^2 , que corresponden con C^1, C^2 transformados por la pose actual del robot. Mediante la restricción al movimiento en el plano $x^3 = 0$, se obtiene el sistema A_p, B_p para el cálculo del vector Φ_p , en este caso compuesto por 4 valores. El sistema con un mínimo de 2 puntos tiene solución directa en este caso.

6.6. Análisis de la complejidad numérica del método

Una de las propiedades más interesantes del método propuesto en este capítulo radica en que su complejidad es $\mathcal{O}(N_Y)$ para un número grande de correspondencias. Puesto que los puntos del tetraedro son siempre 4 independientemente de N_Y , el sistema lineal expresado en (6.9) y en concreto la matriz $A^T A \in \mathbb{R}^{12 \times 12}$, siempre es del mismo tamaño. Despreciando la complejidad de obtener los valores singulares de una matriz 12×12 , si N_Y es suficientemente grande, la complejidad del algoritmo tiene como paso dominante el tiempo empleado en solucionar el producto $A^T A$, ya que la matriz $A \in \mathbb{R}^{2N_Y \times 12}$. Dicho producto tiene una complejidad de $\mathcal{O}(N_Y)$ y al ser el paso más costoso el método completo se puede considerar lineal en complejidad.

En el caso de un número reducido de correspondencias, la etapa dominante del algoritmo recae en el cálculo de los valores singulares de la matriz $A^T A$, siendo aún así una tarea que se puede realizar cientos de veces por segundo dada la tecnología actual.

6.7. Efectos del ruido de la medida y el modelo en la estimación de la pose

En esta sección se discute el efecto que tiene el ruido añadido en las medidas, en la solución propuesta para la pose. La presencia de ruido en las medidas afectará sensiblemente no sólo al valor numérico de la pose, sino también a la identificación de los valores singulares nulos del sistema.

6.7.1. Ruido y autovalores de la matriz $A^T A$

El método presentado en esta sección requiere en general la identificación de los valores singulares nulos de la matriz $A^T A$, para, de este modo, identificar su número y aplicar el desarrollo correspondiente. Como se ha comentado, a lo largo del capítulo, dependiendo de la situación, se ha demostrado analíticamente cuantos valores singulares exactamente iguales a cero se espera encontrar. Sin embargo, en los entornos donde las medidas contenidas en el vector Y_k o las coordenadas tridimensionales del modelo se encuentran contaminadas con ruido, los valores singulares no son exactamente nulos, y el algoritmo debe encontrar aquellos que sean suficientemente pequeños. Los umbrales utilizados para discriminar los valores singulares pequeños son en general difíciles de ajustar automáticamente y no garantizan que los autovalores elegidos sean los correctos. Un error en la elección de los autovalores tiene como consecuencia una pose totalmente desplazada, y por tanto la elección de los autovalores correctos se presenta como un problema crucial.

En el capítulo se ha puesto énfasis en las situaciones que dan lugar a distintos autovalores en función del número de cámaras y puntos del modelo. Se propone por tanto identificar los autovalores atendiendo a dicha clasificación, puesto que el origen de las medidas incluidas en Y_k es conocido. Mediante una tabla de búsqueda, el algoritmo averigua el número de autovalores nulos que se espera encontrar, los cuales deberán coincidir con los de valor inferior encontrados realmente en $A^T A$. En la tabla 6.1, se muestra una ordenación de los autovalores de la matriz en función del número correspondencias simples y múltiples que resume y simplifica todas las condiciones citadas en el texto.

En los casos en los que exista cierta ambigüedad en la elección de los valores singulares más pequeños, como ocurriría si poseen un orden de magnitud similar, se obtendrán las posibles combinaciones, eligiendo aquella que menor error de reproyección presenta. Dicha técnica que también es propuesta en [Moreno-Noguer et al., 2007], aunque en dicho caso no se realiza una clasificación e identificación de las configuraciones que dan lugar a un número determinado de valores singulares.

N_c	Nº Corr. Simples N_Y^s	Nº Corr. Dobles N_Y^d	Nº Corr. Totales	Nº Autovalores nulos
1	>5	0	>5	1
1	5	0	5	2
1	4	0	4	4
>1	4	0	4	4
>1	3	1	4	3
>1	2	2	4	2
>1	1	3	4	1
>1	X	Y	X+Y>4	0

Tabla 6.1: Clasificación de los valores singulares de la matriz en función del número y tipo de correspondencias

6.7.2. Propagación del ruido de medidas y el modelo en la estimación

En este apartado se pretende encontrar un método que permita conocer la incertidumbre que se espera obtener en el cálculo de la pose, a partir de la incertidumbre conocida en medidas y puntos del modelo tridimensional. El cálculo de dicha incertidumbre, definida como una matriz de covarianza, permite integrar debidamente el algoritmo planteado dentro de la estructura de inferencia bayesiana del capítulo 7.

El vector de medidas Y_k se considera como un proceso gaussiano $\mathbf{Y}_k$ de media $\hat{Y}_k$ y matriz de covarianza Σ_L . Suponiendo un vector columna $M = ((M^1)^T, \dots, (M^N)^T)^T$ que contiene los N puntos que forman el modelo tridimensional del robot, se le considera también un proceso aleatorio gaussiano $\mathbf{M}$, descrito mediante su media $\hat{M}$ y matriz de covarianza Σ_M .

Se pretende por tanto, a partir de la descripción gaussiana de las medidas y modelo, obtener una estimación, también gaussiana, de la incertidumbre resultante en la solución de la pose, mediante la obtención de su matriz de covarianza Σ_X . La obtención de una estimación precisa de la matriz Σ_X permitirá obtener en el capítulo siguiente un proceso bayesiano de baja complejidad que hará uso del modelo estadístico aquí presentado.

El algoritmo propuesto en este capítulo para la obtención de la pose, requiere del cálculo de magnitudes que dependen de una forma no lineal con Y_k o M y que en muchas ocasiones son obtenidas mediante técnicas iterativas (p. ej. cálculo de valores singulares en la matriz $A^T A$). La propagación de procesos estadísticos en esas circunstancias requiere de metodologías en las que no sea necesario disponer de una descripción analítica de la función sobre la que se quiere propagar el proceso aleatorio.

Como se comentó en el capítulo 3, en la presente tesis se ha abordado el problema de propagación de procesos aleatorios mediante aproximaciones de primer orden que requieren de funciones derivables, como hace uso el filtro de Kalman Extendido. Sin embargo, en este caso se propone acudir explícitamente a una alternativa que no requiere de dichas aproximaciones, como es la “Transformada Unscented”. Esta transformada, detallada en [Van der Merwe et al., 2000], hace uso de puntos de prueba, llamados “Sigma Points” y cuya evaluación en la función de interés, en este caso el proceso propuesto de pose a partir de medidas en el plano imagen, permite obtener una estimación gaussiana del proceso.

Se plantea obtener, a partir de las descripciones de $\mathbf{Y}_k$ y de $\mathbf{M}$, y mediante la transformada Unscented, la descripción del proceso $\mathbf{X}_k$, que representa la pose obtenida mediante el método mPnP planteado en este capítulo.

En el algoritmo 7 se muestra la aplicación de la transformada “Unscented” en el problema planteado. Se considera al proceso completo de cálculo de pose presentado en este capítulo como una función no analítica nombrada f_{PnP} , en la cual, dado un valor concreto del vector Y_k y un valor para M , obtiene un valor de la pose del robot X_k . La transformada Unscented obtiene una caracterización de la media y varianza de la pose y que definen el proceso $\mathbf{X}_k$. Se considera en todo momento que los procesos $\mathbf{Y}_k$ y $\mathbf{M}$ son incorrelados y por tanto en el algoritmo 7, el cálculo de los puntos de prueba de la transformada “Unscented” se realiza por separado.

La transformada “Unscented” posee una complejidad dominada por el cálculo de la raíz cuadrada de la matriz de covarianza de los procesos $\mathbf{Y}_k$ y $\mathbf{M}$, que se muestra en la ecuación 6.73. La raíz cuadrada de una matriz de covarianza de N filas se calcula mediante la factorización Cholesky de la misma, la cual posee una complejidad $\mathcal{O}(N^3)$. El método propuesto en el algoritmo 7 posee en un principio una complejidad $\mathcal{O}((2N_Y)^3 + (3N)^3)$, dominada por las factorizaciones de las matrices Σ_Y y Σ_M . Sin embargo, se puede considerar que Σ_Y es constante, diagonal en la mayoría de ocasiones, por lo que su raíz cuadrada puede ser precalculada. La matriz Σ_M generalmente variará con el tiempo en el algoritmo presentado en el capítulo 7. Sin embargo, dada la naturaleza recursiva del filtro de Kalman, la factorización Cholesky admitirá una actualización recursiva con una complejidad $\mathcal{O}(N^2)$, siendo N el número de puntos tridimensionales dentro del conjunto M . Dicha actualización recursiva se presentará en el citado capítulo, considerando que el tiempo de cómputo de la misma es el paso dominante del proceso de obtención del proceso $\mathbf{X}$ aquí presentado.

Con el objetivo de justificar el método propuesto, se ha realizado una simulación de Monte Carlo de la desviación real que se produce en la función f_{PnP} en función del ruido introducido en las medidas. El experimento que se ha realizado consta de un total de $N = 40$ puntos de un modelo tridimensional en un espacio de hasta cuatro cámaras $N_c = 4$, cuya distribución geométrica se muestra en la figura 6.5.

Se propone variar la desviación típica de las medidas en el rango siguiente:

$$\sigma_v = (0,01 \quad 0,1 \quad 1 \quad 10 \quad 100),$$

suponiendo una matriz Σ_L compuesta por el valor σ_v^2 en su diagonal. La varianza asociada al modelo, Σ_M , se introduce diagonal suponiendo una desviación típica individual de $\sigma_m = 2$, que quedará fija en el experimento. La simulación de Monte Carlo se realiza para una población total de $N_M = 10^5$ muestras, comprobando que más iteraciones no modifican apreciablemente la varianza muestral. La simulación se ha dividido para una o varias cámaras, ya que como se mostrará a continuación existen diferencias evidentes en el comportamiento y linealidad entre ambas.

Cada una de las N_M muestras de Monte Carlo representan una posible medida Y_k de los N puntos en el conjunto de cámaras y un modelo M de N puntos. Cada una de dichas muestras se ha contaminado con el ruido correspondiente a los procesos $\mathbf{Y}_k$ de medida y el proceso $\mathbf{M}$ del modelo. A partir de la función f_{PnP} , se obtienen un conjunto de N_M posibles valores de pose, definida esta como las tres posiciones en el espacio y los

Algoritmo 7 Obtención de $\mathbf{X}_k$ a partir de $\mathbf{Y}_k$ y $\mathbf{M}$

Require: Distribuciones gaussianas $\mathbf{Y}_k = N(\hat{Y}_k, \Sigma_L)$ y $\mathbf{M} = N(M, \Sigma_M)$.

1: Se obtienen dos conjuntos de $N_S = 2N_X + 1$ ($N_X = 2N_Y + 3N$) pesos escalares Ψ^m y Ψ^Σ :

$$\Psi_0^m = \frac{\lambda}{N_S + \lambda} \quad \Psi_0^\Sigma = \frac{\lambda}{N_S + \lambda} + (1 - \alpha^2 + \beta) \quad (6.71)$$

$$\Psi_i^m = \Psi_i^\Sigma = \frac{1}{2(N_X + \lambda)}, \quad i = 1, \dots, 2N_X, \quad (6.72)$$

donde $\lambda = N_X(\alpha^2 - 1)$. La constante α determina el esparcimiento de los puntos de prueba alrededor de la media ($10^{-4} < \alpha \leq 1$) y $\beta = 2$ para distribuciones gaussianas.

2: Calcular N_S puntos de prueba óptimos (puntos sigma) (χ^Y, χ^M) y sus correspondientes para los procesos $(\mathbf{Y}_k, \mathbf{M})$.

$$\begin{cases} \chi_0^Y = \hat{Y}_k \\ \chi_i^Y = \chi_0^Y + \sqrt{(N_X + \lambda)\Sigma_{Y_i}} & i = 1, \dots, N_X \\ \chi_i^Y = \chi_0^Y - \sqrt{(N_X + \lambda)\Sigma_{Y_i}} & i = N_X + 1, \dots, 2N_X \end{cases} \quad (6.73)$$

$$\begin{cases} \chi_0^M = \hat{M} \\ \chi_i^M = \chi_0^M + \sqrt{(N_X + \lambda)\Sigma_{M_i}} & i = 1, \dots, N_X \\ \chi_i^M = \chi_0^M - \sqrt{(N_X + \lambda)\Sigma_{M_i}} & i = N_X + 1, \dots, 2N_X \end{cases}, \quad (6.74)$$

donde cada término $\sqrt{(N_X + \lambda)\Sigma_{M_i}}$ corresponde con la fila i de la raíz cuadrada de la matriz $(N_X + \lambda)\Sigma_M$.

3: Propagación de los puntos sigma a través de la función f_{PnP} .

$$\chi_i^X = f_{PnP}(\chi_i^Y, \chi_i^M) \quad i = 1..N_S$$

4: Cálculo de la media y varianza del proceso $\mathbf{X}_k$

$$\begin{aligned} \hat{X}_k &= \sum_{i=0}^{N_S} \psi_i \chi_i^X \\ \Sigma_X &= \sum_{i=0}^{N_S} \psi_i [(\chi_i^X - \hat{X}_k)(\chi_i^X - \hat{X}_k)^T] \end{aligned} \quad (6.75)$$

5: La distribución gaussiana buscada para la pose y estructura es $\mathbf{X}_k = (\hat{X}_k, \Sigma_X)$

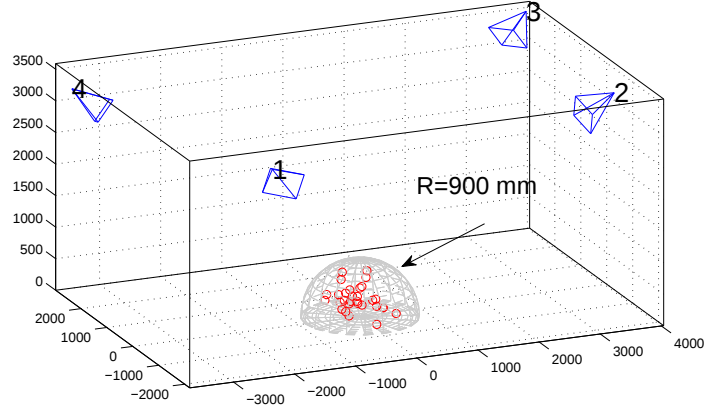


Figura 6.5: Distribución de cámaras y puntos del modelo tridimensional

valores del cuaternión de rotación. Mediante los valores evaluados de pose se obtiene la correspondiente varianza muestral Σ_X^m y media muestral X_k^m del proceso. Dicha varianza y media se compara con aquella obtenida mediante el algoritmo 7. En las figuras 6.6.a y 6.6.b se muestran respectivamente los errores relativos cometidos en la varianza y media entre la simulación de Monte Carlo y el método propuesto, para $N_c = 2$ y $N_c = 1$ cámaras.

El error relativo de la varianza y la media se definen a continuación:

$$\epsilon_\Sigma = \frac{\|\Sigma_X^m - \Sigma_X\|}{\|\Sigma_X\|} \quad \epsilon_X = \frac{\|X_K^m - \hat{X}\|}{\|\hat{X}\|}, \quad (6.76)$$

definiendo el operador norma a una matriz como la norma de todos sus elementos ordenados en un vector columna.

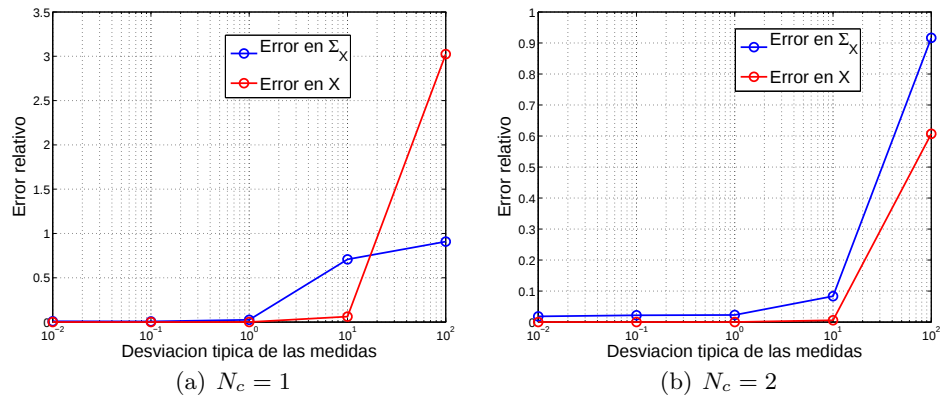


Figura 6.6: Errores relativos en la media y varianza del proceso para una y dos cámaras.

Como se aprecia en la figura 6.6.a, la estimación de la varianza es muy precisa, $\epsilon_\Sigma < 1\%$, para desviaciones típicas inferiores a $\sigma_v = 10$. Superando ese valor, el error en la

estimación aumenta considerablemente. En la figura 6.7 se muestra la estimación de la varianza gráficamente. En las figuras 6.7.a, 6.7.b y 6.7.c se muestran los planos $X - Y$, $X - Z$ e $Y - Z$ respectivamente. En cada uno se han representado como puntos la posición de una porción de las partículas utilizadas en la simulación de Monte Carlo, y como elipses las covarianzas, tanto la muestral como la estimada para el caso en el que $\sigma_v = 0,1$. En las figuras 6.7.d, 6.7.e y 6.7.f se muestran las mismas magnitudes para el caso en el que $\sigma_v = 100$. Como se observa fácilmente, las distribuciones de puntos no son gaussianas en este último caso, y por tanto la transformada Unscented falla en su estimación. Sin embargo para una varianza en el ruido de medida inferior, la distribución es gaussiana y la transformada Unscented es precisa.

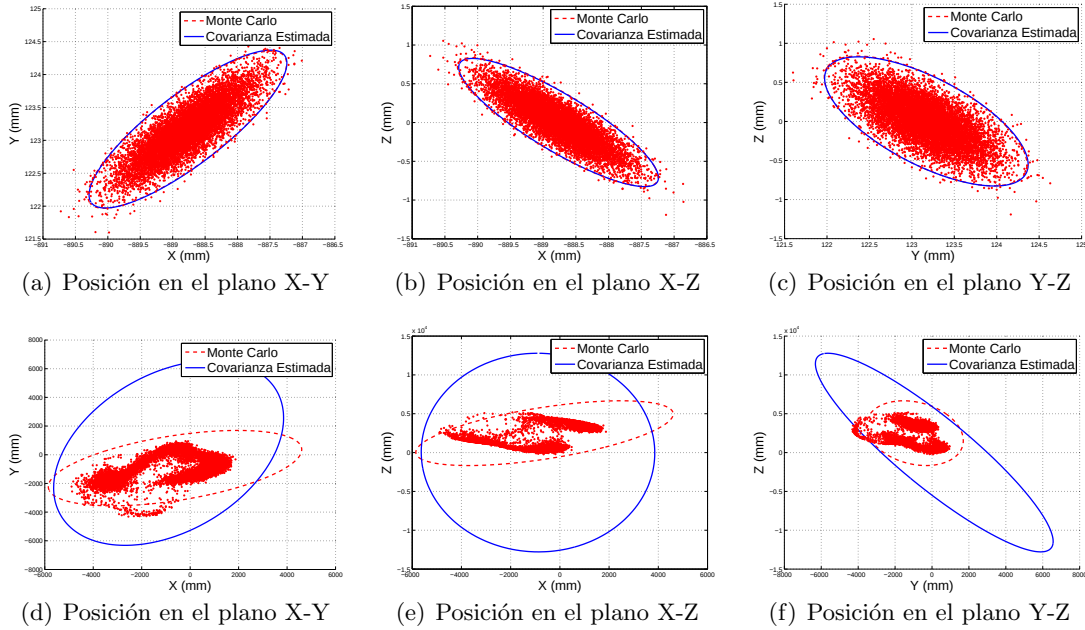


Figura 6.7: Simulación de Monte Carlo de la varianza del error cometido en el cálculo de la pose.

Mediante múltiples cámaras, la estimación de Σ_X es más precisa para un nivel de ruido superior que en el caso de una sola cámara, llegando a ser precisa para una varianza en la imagen de $\sigma_v = 10$. Éste hecho es razonable, ya que en general, mediante varias cámaras el problema está mejor condicionado y por lo tanto es esperable una mejoría, la cual se hará patente también en los resultados que se presentan en la próxima sección.

6.8. Evaluación experimental

Se propone realizar, mediante un entorno de medida simulado, una serie de experimentos que verifiquen y validen el algoritmo propuesto. El entorno simulado que se utilizará en esta sección consta de un conjunto de cámaras dispuestas en la misma distribución geométrica y parámetros de calibración utilizados con cámaras reales. Al igual que en el experimento anterior, la geometría del robot consta de una nube aleatoria de puntos que se encuentra contenida en una esfera de radio $R = 900mm$. La disposición geométrica de las cámaras se

muestra en la figura 6.5. Los parámetros de calibración de las cámaras y más detalles sobre esta disposición se mostrarán en el capítulo 8. La posición en el plano imagen dada por Y_k está contaminada con un ruido aleatorio con desviación típica σ_v que es independiente por cada componente del vector.

El error obtenido en el cálculo de la pose se divide entre aquel cometido en la obtención de la orientación del objeto y aquel derivado del cálculo de la posición. El error en ambos casos se muestra en porcentaje, según la siguiente expresión:

$$\epsilon_T = \frac{\|T_1 - T_2\|}{\|T_2\|} \quad \epsilon_R = \frac{\|q_1 - q_2\|}{\|q_2\|}, \quad (6.77)$$

Donde T_1 y T_2 corresponden con los vectores de posición real y estimado, respectivamente. Los vectores q_1 y q_2 representan a su vez los cuaterniones de orientación real y estimado, obtenidos a partir de las matrices de rotación correspondientes.

Se han propuesto los siguientes experimentos con el objetivo de validar la propuesta presentada para el cálculo de pose mediante múltiples cámaras:

- Influencia del número de correspondencias simples y el ruido de medida σ_v para la estimación de la pose mediante una cámara. (Ver figura 6.8)
- Influencia del número de correspondencias dobles o superiores y el ruido de medida σ_v en la estimación de la pose mediante múltiples cámaras. (Ver figura 6.9)
- Influencia del número de correspondencias y el ruido de medida σ_v en la estimación de la pose en un plano mediante una o varias cámaras. (Ver figura 6.10 y 6.11)

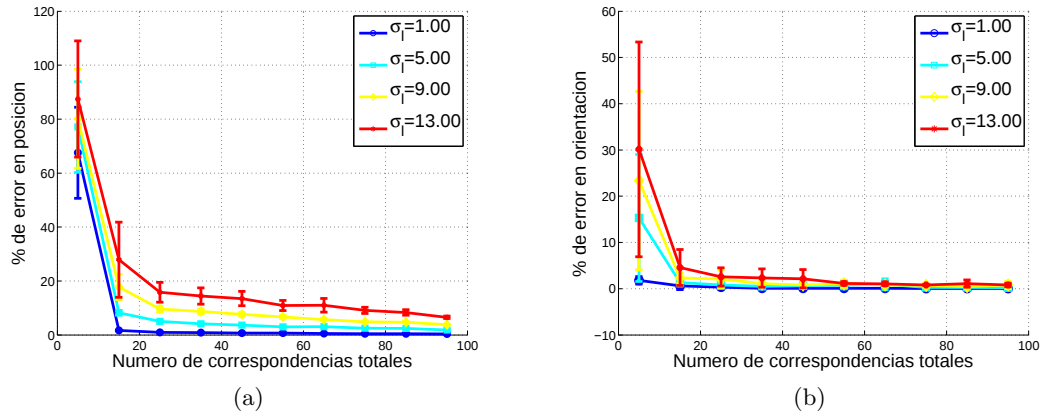


Figura 6.8: Error vs número de correspondencias y varianza del ruido σ_v en la versión de una cámara.

Como se observa en las figuras, las soluciones que implican el uso de una cámara presentan un error relativo considerablemente mayor que el mismo experimento con varias cámaras. La incertidumbre, vista como barras verticales, es reducida en todos los experimentos, salvo en aquellos casos en los que el número de correspondencias es pequeño, en el

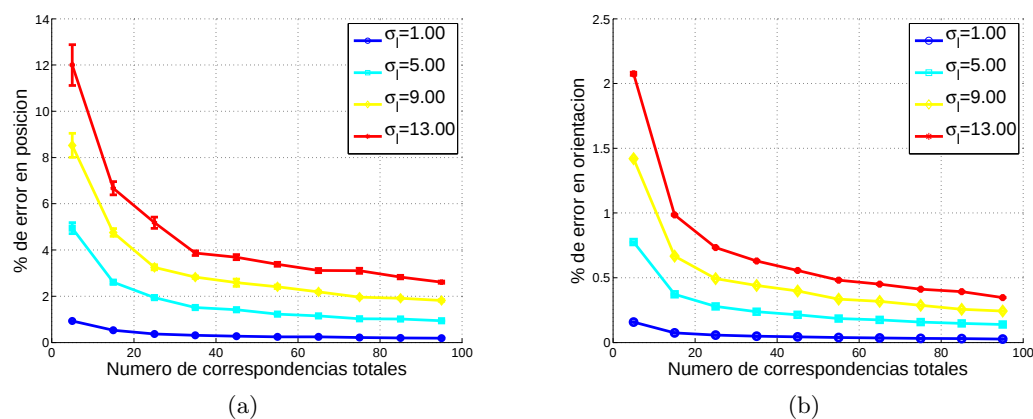


Figura 6.9: Error vs número de correspondencias y varianza del ruido σ_v en la versión de múltiples cámaras.

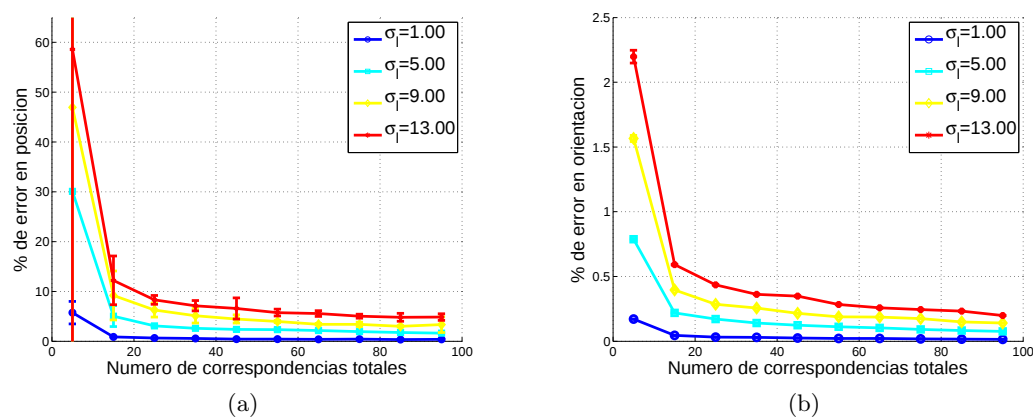


Figura 6.10: Error vs número de correspondencias y varianza del ruido σ_v en la versión de una cámara para un movimiento en el plano X-Y.

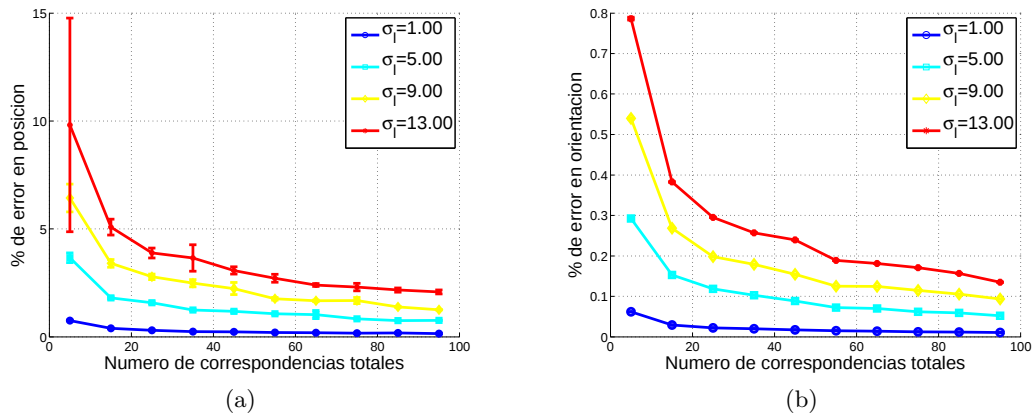


Figura 6.11: Error Vs número de correspondencias y varianza del ruido σ_v en la versión de múltiples cámaras para un movimiento en el plano X-Y.

cual aumenta considerablemente. Se observa también como el método que tiene en cuenta el movimiento en un plano es más preciso que el problema general, y por lo tanto el método se beneficia de la reducción en la dimensionalidad del problema. Para prácticamente todas las simulaciones con un número elevado de puntos, el error se mantiene por debajo del 5 % y por lo tanto se puede considerar un algoritmo preciso.

6.9. Conclusiones

Se ha planteado en este capítulo una solución exacta, no iterativa y de baja complejidad computacional del problema PnP aplicado a múltiples cámaras. El método se basa en el trabajo propuesto por [Moreno-Noguer et al., 2007], el cual, mediante las modificaciones propuestas en este capítulo, permite su aplicación a múltiples cámaras. Se han planteado soluciones tanto para la obtención de la pose tridimensional de un objeto como aquellas en las que el movimiento del mismo se restringe a un solo plano. El problema planteado demuestra ser muy preciso incluso con niveles de ruido altos en el plano imagen. Con el objetivo de conectar el método planteado en este capítulo con el método secuencial presentado en el capítulo 7, se ha propuesto un método que permita obtener una estimación precisa de la incertidumbre obtenida en el resultado de la pose a partir de aquellas en las medidas y el modelo tridimensional del robot.

A continuación se enumeran las aportaciones principales del presente capítulo:

- Se ha ampliado el trabajo de [Moreno-Noguer et al., 2007], que soluciona el problema PnP en una cámara, para su utilización en múltiples cámaras o equivalentemente en cámaras con múltiples centros de proyección. El algoritmo propuesto sigue siendo de complejidad lineal con el número de correspondencias en el plano imagen.
- Se ha propuesto a su vez una solución para movimientos restringidos a un plano y se ha demostrado el incremento en precisión que aporta, además de permitir soluciones

para el problema de los 2 puntos.

- Se ha demostrado analíticamente la relación entre las diferentes configuraciones de correspondencias y los autovalores nulos de la matriz $A^T A$, lo cual permite evitar una detección de los mismos por umbral.
- Se ha propuesto un método que permite evaluar y estimar la incertidumbre que existe en la solución de la pose, dada la incertidumbre en las medidas y el modelo tridimensional.

Capítulo 7

Localización secuencial de robots móviles

7.1. Introducción

En el presente capítulo se aborda el problema de la localización secuencial del robot, esto es, la obtención en cada instante de tiempo k , del valor del vector de estados X_k^a dada la solución del instante anterior X_{k-1}^a y el vector de medidas Y_k . A diferencia del problema planteado en el capítulo anterior, donde la pose del robot se obtiene resolviendo el problema directo a partir de correspondencias de múltiples cámaras, en este caso se propone un esquema de filtrado, en el cual se suponen pequeños movimientos del robot y por tanto se simplifican tanto el proceso de obtención de correspondencias como el propio cálculo de la pose.

A pesar de que la solución del problema mPnP, propuesta en el capítulo anterior, resuelve el problema de la pose sin necesidad de conocer una estimación de la misma en instantes adyacentes, el método de filtrado propuesto en este capítulo posee ventajas adicionales:

- El conocimiento de la pose anterior y su inclusión en un esquema de filtrado permite condicionar la solución obtenida para la pose en el instante siguiente, lo que incrementa la estabilidad de la estimación.
- El filtrado permite simplificar enormemente el problema de correspondencia de puntos en la imagen, ya que limita en gran medida, aunque no completamente, la probabilidad de introducir “outliers” en la misma.
- El esquema propuesto permite aplicar un estimador bayesiano, lo que simplifica la integración coherente de todas las fuentes de información, controlando al mismo tiempo la incertidumbre obtenida en la pose.

7.1.1. Descripción de la propuesta de obtención secuencial de la pose

El proceso “online” planteado, parte inicialmente del instante $k = 0$, utilizando para ello el valor de la pose y la geometría del robot $X_0^a = \{X_0, M^1, \dots, M^N\}$ obtenidos por el proceso de inicialización comentado en el capítulo 5.

Las fuentes de información involucradas en la obtención de X_k^a son las siguientes:

- Estimación de movimiento, proporcionada por la odometría, y la suposición de rigidez de la geometría del robot.

$$X_k^a = f^a(X_{k-1}^a, U_k) = (f(X_{k-1}, U_k), M^1, \dots, M^N) \quad (7.1)$$

- Vector de medidas Y_k compuesto por la proyección $y_k^{i,n}$ de cada punto M^i , $i = 1, \dots, N$, pertenecientes a la geometría del robot, en el conjunto de cámaras disponibles $n = 1, \dots, N_c$.

$$Y_k = \begin{pmatrix} y_k^{1,1} \\ \vdots \\ y_k^{N,1} \\ \vdots \\ y_k^{N,N_c} \end{pmatrix} \quad y_k^{i,n} = h(t(X_k, M^i), P_n) \quad (7.2)$$

El método que se propone en este capítulo es aplicable, independientemente del número de cámaras involucradas y por tanto el vector Y_k posee una descripción genérica.

El hecho de que la solución en el instante anterior X_{k-1}^a sea conocida, junto con la suposición de pequeños incrementos de pose entre cada instante de tiempo, permiten desarrollar un esquema de estimación-corrección.

Como ha sido descrito en el capítulo 3, y esencialmente a lo largo de la tesis, cada proceso dinámico involucrado en el cálculo de la pose es considerado una variable aleatoria, descrita mediante una distribución gaussiana. El problema principal, consistente en obtener el vector de estados X_k^a , dadas las observaciones Y_k y el vector X_{k-1}^a , es considerado como un problema de inferencia estadística. Se propone como objetivo obtener la distribución de probabilidad “a posteriori” del vector de estados dadas las medidas $p(X_k^a | Y_1, \dots, Y_k)$. Dicha distribución representa la estimación de máxima verosimilitud de la solución X_k^a dado el conocimiento e información obtenida hasta el tiempo presente. Mediante la regla de Bayes, se plantea una solución secuencial al problema, la cual es generalizada para cualquier configuración y número de cámaras.

En este capítulo se plantea a su vez la inclusión en el proceso de inferencia de un algoritmo robusto, capaz de eliminar las falsas correspondencias o “outliers” que contaminan el vector de medidas. La discriminación de los “outliers”, al igual que ocurrió en los procesos de inicialización, representa una tarea imprescindible para dotar de estabilidad a las estimaciones realizadas.

El algoritmo “online” planteado se resume por tanto en tres fases principales (ver figura 7.1):

- **Fase de estimación:** la fase de estimación tiene como objetivo obtener la distribución $p(X_k^a | Y_1, \dots, Y_{k-1})$, es decir, la predicción más probable del vector X_k^a , utilizando para ello el modelo de movimiento disponible.
- **Eliminación de “outliers”:** como se ha comentado en anteriores capítulos, el mecanismo de correspondencia entre las medidas en la imagen y los puntos de la geometría del robot, puede introducir “outliers” o falsas correspondencias que contaminan el vector Y_k . En este punto se propone el uso de un mecanismo robusto de detección y eliminación de dichos “outliers”.
- **Fase de Corrección:** Utilizando como vector de medidas actualizado aquel libre de “outliers” obtenido en la fase anterior, la corrección consiste en la obtención de la distribución “a posteriori” $p(X_k^a | Y_1, \dots, Y_k)$ a partir de la obtenida en la fase de estimación.

Tanto el proceso de estimación como el de corrección son resueltos mediante el teorema de Bayes aplicado al problema secuencial, que será descrito convenientemente en el siguiente punto.

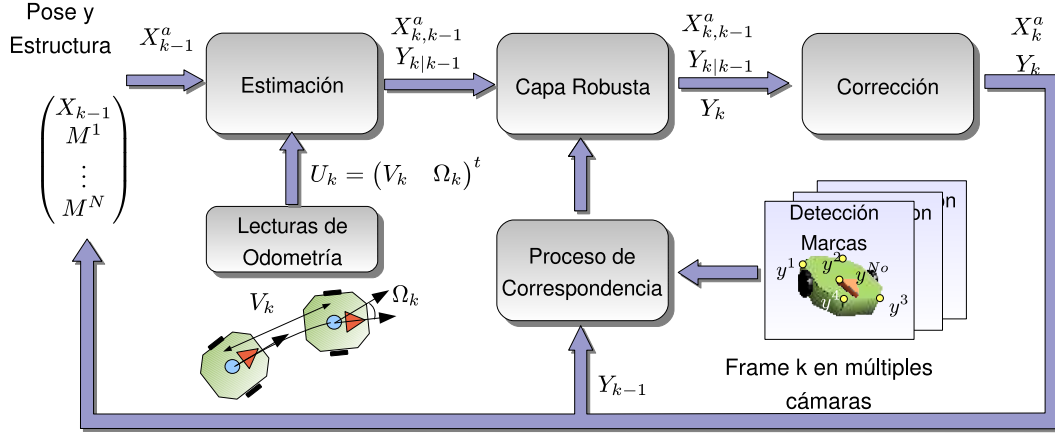


Figura 7.1: Esquema resumido del proceso secuencial para la estimación de la pose

7.2. Proceso de inferencia secuencial

El proceso de inferencia planteado en esta tesis tiene como objetivo la obtención de la distribución “a posteriori” $p(X_k^a|Y_1, \dots, Y_k)$ del vector de estados X_k^a dadas todas las medidas realizadas hasta el momento actual. Se propone el uso del teorema de Bayes como una formalización del algoritmo secuencial, dando lugar a un esquema de estimación y corrección.

Considerando las distribuciones de probabilidad para los procesos implicados, el algoritmo secuencial se compone de la siguiente secuencia:

Dada la distribución “a posteriori” en el instante $k - 1$,

$$p(X_{k-1}^a|Y_1, \dots, Y_{k-1}), \quad (7.3)$$

el proceso de estimación consiste en la obtención de la distribución $p(X_k^a|Y_1, \dots, Y_{k-1})$, la cual representa la mejor estimación del proceso descrito por X_k^a , sin considerar la medida actual Y_k . Esta fase se puede considerar de predicción, puesto que se determina el valor esperado del proceso sin contrastar su estado actual con las medidas. El cálculo de la distribución $p(X_k^a|Y_1, \dots, Y_{k-1})$ implica la resolución de la siguiente ecuación integral:

$$p(X_k^a|Y_1, \dots, Y_{k-1}) = \int_{X_{k-1}^a} p(X_k^a|X_{k-1}^a)p(X_{k-1}^a|Y_1, \dots, Y_{k-1})dX_{k-1}^a. \quad (7.4)$$

La distribución de probabilidad $p(X_k^a|X_{k-1}^a)$ representa el núcleo o distribución de transición de estados del proceso X_k^a , siendo éste considerado por tanto un proceso de Markov. La distribución de transición se modela a partir de los modelos de movimiento comentados en el capítulo 3.

Obtenida $p(X_k^a|Y_1, \dots, Y_{k-1})$, resultado de la fase de estimación, el Teorema de Bayes permite utilizar el vector Y_k para actualizar o corregir la estimación, obteniendo de este modo la distribución objetivo $p(X_k^a|Y_1, \dots, Y_k)$ mediante la regla de Bayes:

$$p(X_k|Y_1, \dots, Y_k) = \frac{1}{h} p(Y_k|X_k^a) p(X_k^a|Y_1, \dots, Y_{k-1}), \quad (7.5)$$

siendo h un coeficiente de normalización necesario para que la función (7.5) sea una función de densidad de probabilidad:

$$h = \int_{X_k^a} p(Y_k|X_k^a) p(X_k^a|Y_1, \dots, Y_{k-1}) dX_k^a \quad (7.6)$$

El algoritmo de inferencia secuencial planteado requiere del cálculo de integrales de funciones de densidad de probabilidad, lo cual da lugar en general a soluciones no analíticas. Se requiere por tanto recurrir a versiones aproximadas de las distribuciones involucradas en la obtención de $p(X_k^a|Y_1, \dots, Y_k)$. En esta tesis se plantea el uso de descripciones gaussianas de cada proceso y por tanto el proceso de inferencia secuencial debe ser particularizado y aproximado por este hecho. Como ya se ha comentado a lo largo de la tesis, los dos primeros momentos de una distribución, (media y varianza) otorgan en la mayoría de los casos la información necesaria para un sistema de localización. Una función gaussiana es definida por completo por ambas magnitudes, siendo identificada la media de la distribución con el valor del proceso utilizado como solución y la varianza con la incertidumbre presente en la misma.

La propagación de procesos gaussianos a través de un proceso de inferencia secuencial es conocido como filtro de Kalman, y su solución es analítica y Gaussiana en ambas fases de estimación y posterior corrección. La aplicabilidad del filtro de Kalman como solución al proceso de inferencia planteado dependerá de la exactitud de las aproximaciones gaussianas de las distribuciones. En general, aún considerando a la distribución $p(X_{k-1}^a|Y_1, \dots, Y_{k-1})$ un proceso gaussiano exacto, el hecho de que el modelo de movimiento y proyección de la cámara, sean funciones no lineales, produce una solución no gaussiana como función de densidad de probabilidad “a posteriori”.

En este capítulo se tratará en detalle el proceso de inferencia cuando se realizan aproximaciones de primer orden de las funciones no lineales involucradas (h^a y f^a) en torno a la media de cada proceso. Las aproximaciones de primer orden permiten obtener una solución compacta de los procesos de estimación y corrección del filtro de Kalman, lo que facilita la obtención de conclusiones y la lectura del algoritmo. El filtro de Kalman en este caso se conoce como el filtro de Kalman extendido (EKF), y ha sido utilizado en procesos de similar complejidad y carácter no lineal que el planteado (ver [Davison et al., 2007]).

7.3. Método “online” de cómputo de pose a partir de múltiples cámaras

Teniendo en cuenta que los procesos deberán ser gaussianos, la distribución obtenida en el frame anterior $p(X_{k-1}^a | Y_1, \dots, Y_{k-1})$ se describe mediante su vector media $\hat{X}_{k-1}^a$ y su matriz de covarianza Σ_{k-1}^a . La distribución inicial, $N(X_0^a, \Sigma_0^a)$ es obtenida mediante el proceso de inicialización comentado en el capítulo 5.

La distribución “a posteriori”, $p(X_k^a | Y_1, \dots, Y_k)$ se obtiene en dos fases de filtrado: fase de estimación y fase de corrección.

7.3.1. Fase de estimación

La fase de estimación hace uso de los modelos de movimiento disponibles con el objetivo de inferir o predecir la próxima pose del robot. El uso de modelos de movimiento se describe en numerosas ocasiones como un proceso de acumulación de incertidumbre, como fue comentado en el capítulo 3, en el que se muestra como la covarianza de la posición aumenta con la longitud del camino recorrido si se utilizan medidas de velocidad. Partiendo de la distribución $p(X_{k-1}^a | Y_1, \dots, Y_{k-1}) = N(\hat{X}_{k-1}^a, \Sigma_{k-1}^a)$, los modelos de movimiento y el ruido incluido en la odometría, se obtiene la siguiente expresión:

$$\begin{aligned}\hat{X}_{k|k-1}^a &= g^a(\hat{X}_k^a, U_k) \\ \Sigma_{k|k-1}^a &= J_x^T \Sigma_{k-1}^a J_x + J_U^T \Sigma_W J_U,\end{aligned}\tag{7.7}$$

siendo $\hat{X}_{k|k-1}^a$ y $\Sigma_{k|k-1}^a$ la media y varianza de la función $p(X_k^a | Y_1, \dots, Y_{k-1})$. Las matrices J_x y J_U representan los Jacobianos de la función g^a con respecto a X_k^a y U_k respectivamente. Generalmente en sistemas de posicionamiento por odometría, la matriz J_x es la identidad, de modo que la matriz de covarianza resultante $\Sigma_{k|k-1}^a$ posee autovalores mayores que Σ_{k-1}^a , lo cual indica el incremento en incertidumbre que supone la fase de estimación.

7.3.2. Fase de corrección

La fase de corrección obtiene la distribución “a posteriori”, usando para ello el vector de medidas Y_k actualizado y la distribución obtenida durante la fase de estimación. La transformación de $p(X_k^a | Y_1, \dots, Y_{k-1})$ en la distribución buscada $p(X_k^a | Y_1, \dots, Y_k)$ requiere de la obtención de $p(Y_k | X_k^a)$, obtenida a partir de los modelos de proyección en el plano imagen y se define mediante el vector media $\hat{Y}_{k|k-1}$ y la matriz de covarianza Σ_k^a :

$$\begin{aligned}\hat{Y}_{k|k-1} &= h^a(X_k^a) \\ \Sigma_{Y_{k|k-1}} &= J_h^T \Sigma_{k|k-1}^a J_h + \Sigma_V,\end{aligned}\tag{7.8}$$

donde J_h es el Jacobiano de la función h^a con respecto a X_k^a y Σ_V es la matriz de covarianza asociada al ruido de medida y comentado en el capítulo 3.

El proceso de corrección incluido por el filtro de Kalman establece una corrección lineal de los procesos $X_{k|k-1}^a$ y $\Sigma_{k|k-1}^a$ por medio de la matriz de Kalman K_G :

$$\begin{aligned} K_G &= \Sigma_{k|k-1}^a J_h (\Sigma_{Y_{k|k-1}})^{-1} \\ X_k^a &= X_{k|k-1}^a + K_G (Y_k - Y_{k|k-1}) \\ \Sigma_k^a &= (I - K_G J_h^T) \Sigma_{k|k-1}^a \end{aligned} \quad (7.9)$$

7.3.3. Complejidad algorítmica

El algoritmo secuencial propuesto, que hace uso del filtro de Kalman y utiliza como vector de medidas el propio Y_k , posee una complejidad de orden $\mathcal{O}(N_Y^3)$, suponiendo que $2N_Y \geq 3N$. Dicha complejidad corresponde al paso dominante que consiste en la inversión de la matriz $\Sigma_{Y_{k|k-1}}$, necesaria para el cómputo de la ganancia de Kalman K_G . La matriz $\Sigma_{Y_{k|k-1}}$, puesto que las medidas están correladas, es de estructura densa y por tanto no se puede proponer un método de cálculo rápido de su inversa.

En general, cuando el número de puntos pertenecientes al modelo tridimensional del robot es grande y hay un número de cámaras suficientemente elevado para que todos los puntos sean observados al menos por una de ellas, la complejidad del filtro de Kalman es considerable, pudiendo comprometer una implementación en tiempo real o soluciones al problema de bajo coste.

7.4. Método secuencial de cálculo de pose de baja complejidad

Se propone en esta sección una solución que reduce la complejidad del algoritmo “online” planteado, convirtiéndolo en un problema de complejidad $\mathcal{O}(N^2)$, es decir, cuadrática con el número de puntos en el modelo tridimensional del robot. La idea básica consiste en transformar el proceso de medida $\mathbf{Y}_k$, que se forma a partir de medidas en el plano imagen, en el proceso $\mathbf{S}_k$, que representa directamente la pose del robot. La transformación entre ambos procesos se obtiene mediante el algoritmo propuesto en el capítulo anterior, el cual como se detalló, permite obtener la pose a partir del modelo tridimensional del robot y un conjunto de medidas del mismo con una baja carga computacional.

La utilización del proceso $\mathbf{S}_k$, en sustitución de $\mathbf{Y}_k$, reduce el tamaño de las medidas N_r , al del vector que compone la pose (p.ej. $N_r = 3$ en el caso de un robot en un plano), disminuyendo la complejidad del filtro de $\mathcal{O}(N_Y^3)$ a cuadrática con el número de puntos del modelo tridimensional. En la figura 7.2 se muestra un esquema del algoritmo planteado, y los procesos involucrados.

A continuación se detallan sus diferentes fases de estimación y corrección.

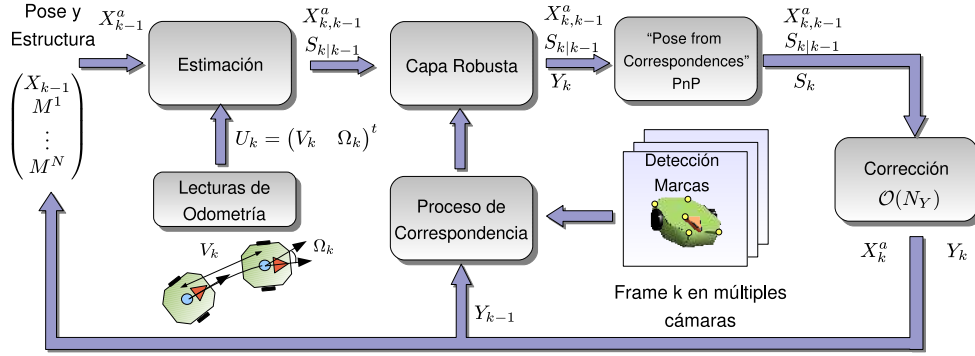


Figura 7.2: Diagrama del proceso online de baja complejidad algorítmica

7.4.1. Vector de medidas

El proceso de medida $\mathbf{S}_k$ se obtiene a partir de los procesos $\mathbf{M}$, formado a partir del conjunto de puntos de la estructura del robot contenidos en $\mathbf{X}_{k-1}^a$, y el proceso de medida en el plano imagen $\mathbf{Y}_k$. Utilizando el algoritmo 7, comentado en el capítulo 6, se obtiene una caracterización estadística de $\mathbf{S}_k$, descrita mediante un vector media $\hat{S}_k$ y una matriz de covarianza que denominaremos Σ_{S_k} .

El vector S_k corresponderá con X_k que determina la pose del robot y que ocupa los primeros elementos del vector de estados X_k^a . De este modo, el nuevo vector de medidas, S_k , se relaciona con el vector de estados completo mediante la siguiente expresión:

$$S_k = C X_k^a + V_{S_k} \quad C = (I_{N_r \times N_r}, \bar{0}_{N_r \times 3N}), \quad (7.10)$$

El cálculo de Σ_{S_k} depende de la configuración de medidas en el vector Y_k y la covarianza de éstas Σ_V , así como de la covarianza $\Sigma_{M,k}$ asociada a los puntos del modelo M observados. El cálculo de Σ_{S_k} se ha desarrollado en el capítulo 6, utilizando para ello la transformada “Unscented”. Como se comentó en dicho capítulo, el cálculo de Σ_{S_k} requiere en general una complejidad $\mathcal{O}(N^3)$, dada por el cálculo de la factorización Cholesky de la matriz $\Sigma_{M,k}$, suponiendo el precálculo de la factorización de Σ_V . En el siguiente apartado se comentará cómo obtener de manera recursiva dicha factorización, lo cual reducirá su complejidad a $\mathcal{O}(N^2)$.

7.4.2. Etapas de estimación y corrección

La principal diferencia en el método propuesto de baja complejidad, con respecto al propuesto en la sección anterior, es el vector de medidas, y por tanto, la estimación $X_{k|k-1}^a$ permanece igual a la anteriormente propuesta. Partiendo de la distribución $p(X_{k-1}^a | S_1, \dots, S_{k-1}) = N(\hat{X}_{k-1}^a, \Sigma_{k-1}^a)$, los modelos de movimiento y el ruido incluido en la odometría, se obtiene la distribución expuesta en la expresión (7.7).

La etapa de corrección sin embargo se simplifica considerablemente como se detalla a continuación:

La fase de corrección elimina la incertidumbre añadida en la estimación usando para ello el vector de medidas S_k actualizado y calculado mediante el método planteado en el capítulo anterior. La transformación de la distribución $p(X_k^a|S_1, \dots, S_{k-1})$ en la distribución buscada $p(X_k^a|S_1, \dots, S_k)$ requiere de la obtención de la distribución $p(S_k|X_k^a)$, la cual se obtiene a partir de la matriz C .

$$\begin{aligned}\hat{S}_{k|k-1} &= CX_{k|k-1}^a \\ \Sigma_{S_{k|k-1}} &= C^T \Sigma_{k|k-1}^a C + \Sigma_{VS_k},\end{aligned}\quad (7.11)$$

donde C en realidad es una matriz que trunca el vector X_k^a en los términos que definen la pose X_k . La expresión (7.11) es exacta y no requiere del cálculo de derivadas aunque sí el cálculo de Σ_{VS_k} mediante la transformada “Unscented”.

El proceso de corrección incluido por el filtro de Kalman establece una ponderación lineal de los procesos $X_{k|k-1}^a$ y $\Sigma_{k|k-1}^a$ por medio de la matriz de Kalman K_G :

$$\begin{aligned}K_G &= \Sigma_{k|k-1}^a C (\Sigma_{S_{k|k-1}})^{-1} \\ X_k^a &= X_{k|k-1}^a + K_G (S_k - S_{k|k-1}) \\ \Sigma_k^a &= (I - K_G C^T) \Sigma_{k|k-1}^a\end{aligned}\quad (7.12)$$

Puesto que el vector de observación S_k es directamente una parte del vector X_k^a , la mayoría de las operaciones involucradas en el filtro de Kalman se simplifica enormemente, siendo innecesario el uso de Jacobianos en la fase de corrección.

Cálculo recursivo de la factorización Cholesky

Como se ha comentado con anterioridad, es necesario el cálculo de la factorización Cholesky de la matriz $\Sigma_{M,k}$, necesaria para obtener la matriz Σ_{S_k} , que representa la matriz de covarianza del ruido del vector de medidas S_k . La matriz $\Sigma_{M,k}$ consiste en la submatriz Σ_k^a correspondiente a las $3N$ últimas filas y columnas, correspondiendo con la información del modelo tridimensional del robot:

$$\Sigma_{M,k} = C_2 \Sigma_k^a C_2^T \quad C_2 = \begin{pmatrix} \bar{0}_{3N \times 3} & I_{3N \times 3N} \end{pmatrix}. \quad (7.13)$$

Dada la matriz $\Sigma_{M,k}$, simétrica y definida positiva, la factorización Cholesky consiste en una matriz L_k tal que:

$$\Sigma_{M,k} = (L_k^M)^t L_k^M \quad L_k^M = \sqrt{\Sigma_{M,k}} \quad (7.14)$$

donde L_k^M es una matriz triangular. La factorización Cholesky, comúnmente utilizada como raíz cuadrada de una matriz, posee una complejidad $\mathcal{O}(N^3)$, siendo N el número de puntos en el modelo tridimensional del robot, contenidos en vector de estados.

La matriz Σ_k^a se actualiza mediante la ecuación de corrección de Kalman (7.9), pudiéndose aprovechar dicha regla de actualización para calcular con baja complejidad el valor de L_k^M .

Se parte de la expresión (7.12), sustituyendo el valor de K_G :

$$\Sigma_k^a = \Sigma_{k|k-1}^a - \Sigma_{k|k-1}^a C (\Sigma_{S_{k|k-1}})^{-1} C^T \Sigma_{k|k-1}^a, \quad (7.15)$$

donde $\Sigma_{S_{k|k-1}}$ es una matriz del tamaño de la pose (3×3) y $\Sigma_{k|k-1}^a C$ es igual a las tres primeras columnas de la matriz. Aplicando la expresión (7.13) se obtiene la regla recursiva para obtener la matriz $\Sigma_{M,k}$:

$$\Sigma_{M,k} = \Sigma_{M,k|k-1} - C_2 \Sigma_{k|k-1}^a C (\Sigma_{S_{k|k-1}})^{-1} C^T \Sigma_{k|k-1}^a C_2^T \quad (7.16)$$

La expresión (7.16) se simplifica del modo siguiente:

$$\Sigma_{M,k} = \Sigma_{M,k|k-1} - L_S^T L_S \quad L_S^T = C_2 \Sigma_{k|k-1}^a C \sqrt{(\Sigma_{S_{k|k-1}})^{-1}}, \quad (7.17)$$

donde L_S^T es una matriz $3N \times 3$ y $\sqrt{(\Sigma_{S_{k|k-1}})^{-1}}$ es la factorización Cholesky de la inversa de la matriz $\Sigma_{S_{k|k-1}}$, cuyo cálculo, al ser una matriz de reducido tamaño no afecta a la complejidad dominante del problema. De este modo el cálculo de L_S implica una complejidad de orden $\mathcal{O}(3N^2)$.

El término $L_S^T L_S$ se puede expresar como la siguiente suma:

$$L_S^T L_S = L_1^T L_1 + L_2^T L_2 + L_3^T L_3, \quad (7.18)$$

siendo L_1, L_2, L_3 las tres filas que componen la matriz L_S . La actualización recursiva de la covarianza, descrita en (7.17) queda por tanto como sigue:

$$\Sigma_{M,k} = \Sigma_{M,k|k-1} - (L_1^T L_1 + L_2^T L_2 + L_3^T L_3) \quad (7.19)$$

Si se denomina como $L_{k|k-1}^M$ a la factorización de la matriz $\Sigma_{M,k|k-1}$, el objetivo es encontrar un método de complejidad cuadrática con la dimensión del vector de estados que calcule la factorización L_k^M .

Acudiendo al método propuesto en [Dongarra et al., 1979], si L es la factorización Cholesky de la matriz Σ , entonces el cálculo de la factorización Cholesky de la matriz $\Sigma + X^T X$, donde X es un vector fila, se puede realizar con complejidad cuadrática con el número de elementos del vector X . Aplicando dicho método a la ecuación (7.19) de manera recursiva para los tres vectores fila L_1, L_2, L_3 , se obtiene la factorización L_k^M en función de $L_{k|k-1}^M$ con complejidad $\mathcal{O}(N^2)$. El desarrollo propuesto será cierto siempre que $\Sigma_{M,k|k-1}$

sea igual a $\Sigma_{M,k-1}$, lo cual se cumple ya que el modelo tridimensional no influye en las ecuaciones de movimiento del robot.

7.4.3. Complejidad algorítmica

En el caso del método propuesto en esta sección, el paso dominante no recae en el cálculo de la inversa de la covarianza de las medidas, en este caso $\Sigma_{S_{k|k-1}}$, puesto que el tamaño del vector utilizado para representar la pose es reducido. En este caso, la complejidad del algoritmo está dominada por $\mathcal{O}(N^2)$, para el cómputo de la factorización recursiva de la matriz de covarianza $\Sigma_{M,k}$, necesaria para el cálculo del vector S_k .

Se puede por tanto asegurar que el algoritmo propuesto es cuadrático con el número de marcas existentes en el modelo tridimensional, lo cual mejora en al menos un orden de magnitud una propuesta basada en el uso de las medidas en el plano imagen directamente en el filtro de Kalman. Como ventaja añadida se tiene la reducción en el número de derivadas necesarias, siendo tan solo necesarias en el caso del paso de estimación.

7.5. Capa robusta

El vector Y_k utilizado como vector de medidas, extraídas en el plano imagen, es susceptible de contener “outliers”. El proceso “online” propuesto en esta sección producirá una estimación coherente de la pose siempre que Y_k represente una realización concreta del proceso de medida, produciendo en caso contrario una desviación en la estimación, la cual supone divergencia en el cálculo de la pose.

El objetivo de la capa robusta, incluida entre el proceso de estimación y corrección del filtro de Kalman, es la eliminación de los “outliers” presentes en el vector de medidas Y_k , de modo que el filtro no produzca correcciones inconsistentes en el vector de estados. En esta tesis se propone el uso del algoritmo RANSAC (“Random Sample Consensus”), ampliamente utilizado en visión artificial como algoritmo robusto en problemas de similar complejidad que el propuesto en esta tesis.

El funcionamiento de este tipo de algoritmos se basa en la inclusión de restricciones geométricas sobre las componentes del vector de medidas Y_k , que permiten discernir cuales de ellas son consideradas “outliers”. Las restricciones impuestas se diseñan para que un subconjunto cualquiera de Y_k , o conjunto mínimo, genere un modelo geométrico, el cual puede ser utilizado para comparar la cantidad de puntos de Y_k que son consistentes con dicho modelo. Mediante una política estadística de muestreo de subconjuntos de Y_k y comprobación de los modelos obtenidos frente al conjunto de medidas, se averigua hasta un grado de “confianza” requerido la identidad de las medidas descartadas.

La elección del modelo utilizado en el método RANSAC depende en gran medida de las condiciones concretas del problema. En el caso más simple utilizado para describir este tipo de métodos se muestran resultados en la detección robusta de primitivas geométricas como rectas o planos, y en casos más aplicados se utilizan restricciones geométricas entre cámaras como puede ser homografías, la matriz fundamental o el tensor trifocal.

En el caso que ocupa a esta tesis, el hecho de poseer un modelo tridimensional del objeto buscado contenido en X_{k-1}^a , permite elegir como modelo la pose del objeto X_k actual, dadas un conjunto de s medidas.

Como fue desarrollado en el capítulo 6, se dispone de un método que obtiene la pose del objeto dadas las medidas y el modelo tridimensional. Se requiere de un mínimo de $s = 4$ puntos si el robot se mueve en el espacio tridimensional y un mínimo de $s = 2$ si el robot se mueve en el plano horizontal. Como se describirá a continuación, el número mínimo de correspondencias necesarias para obtener la pose estará directamente relacionado con el número de iteraciones necesarias en el método RANSAC, para eliminar “outliers”.

7.5.1. Descripción del algoritmo RANSAC

Se presenta a continuación una breve descripción del algoritmo RANSAC, de manera que sirva de referencia en el texto y se comentarán las alternativas utilizadas para definir las restricciones geométricas del problema.

A continuación se exponen las definiciones necesarias para la descripción del algoritmo RANSAC:

- La probabilidad de aparición de los “outliers” en el vector Y_k se define como p_{out} , y su valor será calculado y actualizado por el propio algoritmo.
- Se define como Y_k^s a un subconjunto de s elementos del vector Y_k elegidos de manera aleatoria. En el caso de que Y_k contenga correspondencias de múltiples cámaras, el muestreo no varía, puesto que el algoritmo desarrollado en el capítulo 6 admite correspondencias de múltiples cámaras.
- Para cada grupo de s elementos existe una función f_{PnP} que calcula el vector de parámetros del modelo Φ , utilizando para ello el conjunto Y_k^s y el conjunto de puntos tridimensionales M , incluidos en el vector de estados obtenido del proceso de estimación $X_{k|k-1}^a$.

$$\Phi = f_{PnP}(Y_k^s, M) \quad (7.20)$$

De manera implícita quedarían ocultos otros parámetros de los que depende la transformación f_{PnP} , como pueden ser parámetros de calibración de las cámaras o matrices de covarianza de los procesos implicados.

- El método RANSAC obtiene, mediante la expresión 7.20, candidatos a la pose del robot, de los cuales requiere averiguar el grado de consenso con las medidas. Para ello se define una función escalar $D(\Phi, y_k^i)$, que representa la distancia de una medida y_k^i al modelo Φ .
- La probabilidad de encontrar al menos un conjunto de s medidas Y_k^s libre de “outliers” se define como p (e.g. $p = 0,99$). Dicha probabilidad actúa a modo de intervalo de confianza del método y es un parámetro de diseño.

- El número de medidas que están de acuerdo (“inliers”) con el modelo obtenido Φ , o de otro modo, que obtienen una distancia D al modelo Φ suficientemente pequeña, se caracteriza por N_{in} . Se denomina ϵ al porcentaje de inliers N_{in} con respecto al total
- El modelo geométrico Φ_{best} corresponde con aquel que obtiene un número de “inliers” mayor N_{in}^{best} .
- El número de iteraciones N^{it} necesario para cumplir con la probabilidad p , suponiendo un muestreo aleatorio, se obtiene con la siguiente expresión:

$$N^{it} = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)}, \quad (7.21)$$

donde el valor de s determina, según la expresión 7.21, el número de iteraciones necesarias para cumplir las restricciones impuestas por p . En el algoritmo 8 se muestra la versión algorítmica del método RANSAC. El funcionamiento básico consiste en encontrar mediante muestreo varios conjuntos Y_k^s de s medidas. Cada uno da lugar a un modelo Φ y un conjunto de “inliers” asociado a ese modelo. Aquel Φ con mayor número de “inliers” es considerado como el mejor candidato libre de “outliers” del vector Y_k .

Algoritmo 8 RANSAC

- 1: $N^{it} = 0$, $N_{in}^{best} = 0$, X_{best}^a , $n = 1$, $p = 0,99$.
 - 2: **while** $n < \min(N^{it}, N_{max}^{it})$ **do**
 - 3: Muestreo aleatorio del vector Y_k para obtener un conjunto Y_k^s con s elementos.
 - 4: Obtención de una hipótesis para la pose $\Phi = f_{PnP}(Y_k^s, X_{k|k-1}^a)$ y el conjunto $Y^{in} = \{y_k^i \mid |D(\Phi, y_k^i)| < d_{max}\}$ con N_{in} .
 - 5: **if** $N_{in} > N_{in}^{best}$ **then**
 - 6: $\Phi_{best} = \Phi$, $N_{in}^{best} = N_{in}$, $\epsilon = N_{in}/N_Y$.
 - 7: $N^{it} = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)}$
 - 8: **end if**
 - 9: $n = n + 1$
 - 10: **end while**
 - 11: La identidad de los N_{in}^{best} representa los “inliers” del vector Y_k .
-

Una vez identificadas aquellas correspondencias del vector Y_k que son consideradas “inliers”, se elimina el resto, truncando a su vez las matrices de covarianza relacionadas con dichas componentes.

7.6. Evaluación experimental del método propuesto

En esta sección se propone un experimento realizado en un entorno simulado, que tiene como objetivo validar el algoritmo “online” planteado en este capítulo. El entorno de prueba utilizado para este experimento coincide con aquel utilizado en el capítulo 5 para presentar los resultados del algoritmo de inicialización con múltiples cámaras.

El entorno de simulación consta por tanto de un total de $N_c = 8$ cámaras, con la disposición mostrada en la figura 5.15 del capítulo 5. Las características del entorno de prueba coinciden con aquellas descritas en la sección 5.3.8 del citado capítulo, con la salvedad del número de puntos utilizados para describir el robot. En este caso se utilizarán un total de $N = 35$ puntos repartidos uniformemente en el interior de un cilindro de radio $0,5m$ y altura $1m$. La trayectoria utilizada para la evaluación experimental es también la misma utilizada en la inicialización, siendo circular de radio $1m$, y mostrándose en la figura 5.15

El error de localización se mostrará como el error de posición ϵ_T y orientación ϵ_α medios a lo largo de la trayectoria. El error instantáneo, correspondiente a la estimación de la pose del robot en un instante concreto de la trayectoria se denotará como ϵ_{T_k} y ϵ_{α_k} .

En todos los experimentos propuestos, se supone que el vector de estados X_k^a es inicializado mediante dos cámaras, utilizando para ello un cuarto de la trayectoria circular. El método “online” presentado completará una circunferencia a partir del punto donde concluye la inicialización.

En todos los experimentos se evalúan ambos métodos presentados en este capítulo. Se denominará como método “cúbico”, a aquel que hace uso de medidas en el plano imagen como vector Y_k en el filtro de Kalman y cuya complejidad es $\mathcal{O}(N_Y^3)$. Se denominará como método “cuadrático” a la aproximación $\mathcal{O}(N^2)$.

Se proponen los siguientes experimentos:

- Comparación del error de localización en función del número de puntos de la estructura del robot. En este experimento se varía el número de puntos que componen el vector de estados en el rango siguiente:

$$N = (2 \quad 5 \quad 10 \quad 15 \quad 20 \quad 25 \quad 30 \quad 35),$$

mostrándose los resultados del error medio de la pose a lo largo de la trayectoria para los dos métodos. En las figuras 7.3.a y 7.3.b se muestran, respectivamente, los errores de posición y orientación para el método “cúbico” y en las figuras 7.4.a y 7.4.b se muestran los errores para el método “cuadrático”.

- Error de localización en función del porcentaje de “outliers” en el vector Y_k . Se varía el valor de p_{out} en el rango:

$$p_{out} \in (0 \quad 0,05 \quad 0,1 \quad 0,15 \quad 0,2),$$

mostrándose los resultados de localización del método “cúbico” en las figuras 7.5.a, 7.5.b, y los del método “cuadrático” en la figuras 7.6.a, 7.6.b.

Como puede observarse en los resultados obtenidos, el método cuadrático posee una precisión inferior a aquella obtenida mediante el uso de Y_k como medidas en el filtro de Kalman. Este hecho pone de manifiesto que el filtro de Kalman en su versión de complejidad “cúbica” aprovecha el hecho de que la pose en dos instantes de tiempo varía muy poco. En el caso del método cuadrático, la pose es obtenida “globalmente” a partir de las medidas, siendo por tanto independiente del desplazamiento entre la estimación de la pose por la odometría y el valor medido. Si el número de correspondencias es pequeño y los desplazamientos del robot son pequeños, dependiendo de la aplicación, puede compensar la utilización del algoritmo más lento.

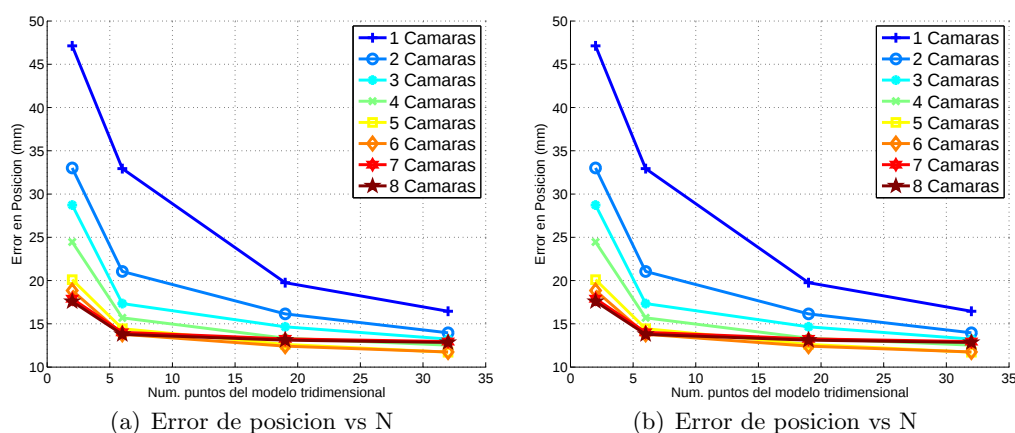


Figura 7.3: Errores de localización en función del número de puntos en la estructura del robot.

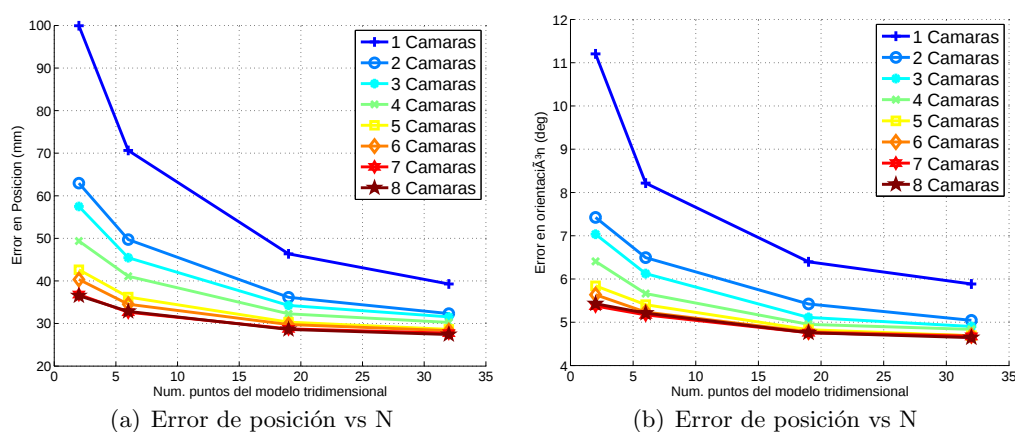


Figura 7.4: Errores de localización en función del número de puntos en la estructura del robot.

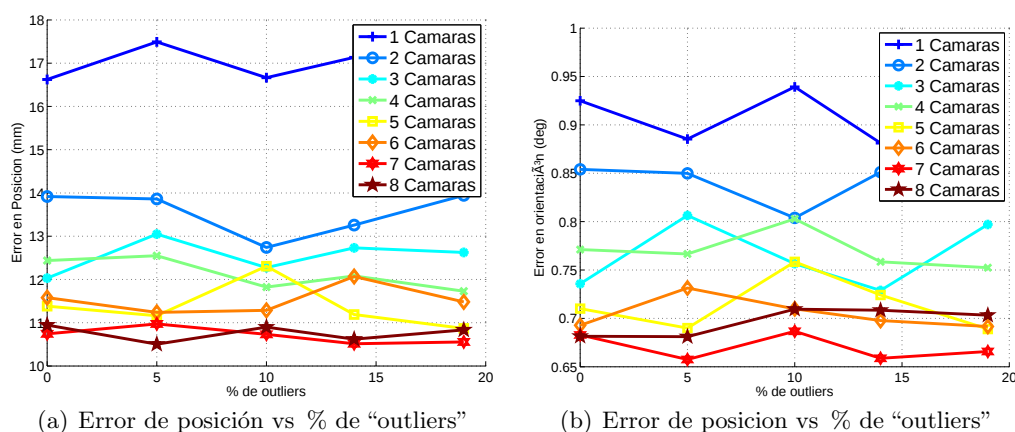


Figura 7.5: Errores de localización en función del porcentaje de "outliers" presente en el vector de medidas.

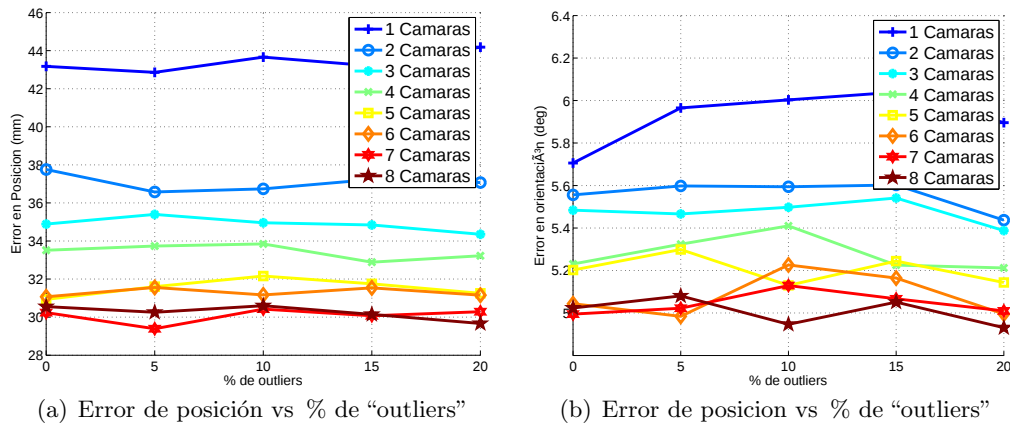


Figura 7.6: Errores de localización en función del porcentaje de “outliers” presente en el vector de medidas, utilizando el método de complejidad cuadrática.

7.7. Inclusión de nuevos puntos en el vector de estados

En la presente sección se muestra la propuesta utilizada en esta tesis para enriquecer el modelo tridimensional, contenido en X_k^a , de manera secuencial. De este modo, el conocimiento acerca del objeto a localizar se incrementa a medida que éste muestra geometría, no reconstruida previamente en el proceso de inicialización. Uno de los requerimientos deseables para el algoritmo propuesto en esta sección es que sea integrado de manera natural en la estructura Bayesiana utilizada para resolver la pose secuencialmente.

Al igual que en el proceso de inicialización, existen diferentes planteamientos según el número de cámaras y su disposición. En esta sección se distinguirán dos casos fundamentales:

- Inclusión de nuevos puntos en el vector de estados utilizando una única cámara.
- Reconstrucción de puntos nuevos utilizando correspondencias entre cámaras.

En cada caso, las técnicas empleadas difieren de manera sustancial. En este capítulo se prestará especial atención al caso que emplea una sola cámara, el cual es significativamente más complejo que su homónimo haciendo uso de múltiples cámaras, y que ha sido amplio objeto de estudio en la literatura (ver [Hartley and Zisserman, 2003]).

7.7.1. Inclusión de geometría en el vector de estados utilizando una única cámara

La propuesta presentada en este punto tiene como objetivo la inclusión de nuevos puntos de la geometría del robot a partir de medidas tomadas mediante una cámara y la información obtenida en cada instante de tiempo por el algoritmo “online” propuesto con anterioridad. El proceso que se detalla a continuación se integra de manera natural

con la obtención de pose y el marco bayesiano de estimación-corrección, ofreciendo como resultado un sistema de localización simultáneo a la reconstrucción del robot.

Se propone realizar un seguimiento o “tracking” de aquellas medidas de la imagen que son detectadas como puntos de interés, pero que no corresponden con ninguno de los puntos tridimensionales contenidos en el vector de estados X_k^a . Dichas marcas o medidas corresponden a puntos nuevos, siendo de sumo interés su integración en el vector de estados.

Se describe por simplicidad el algoritmo que obtiene las coordenadas tridimensionales de un único punto nuevo, que se nombrará como M^o . Dicho punto nuevo es observado por primera vez en $k = k^o$ mediante una medida $y_{k^o}^o$ en el plano imagen de la única cámara utilizada. El algoritmo de “tracking” de marcas naturales obtiene, a partir de $y_{k^o}^o$, y mediante la misma técnica utilizada en el proceso de inicialización del capítulo 5, el conjunto de medidas correspondientes al punto M^o en instantes de tiempo consecutivos $y_{k^o}^o, \dots, y_k^o$ hasta el momento presente.

Desde el punto de vista de la inferencia Bayesiana, el objetivo que se busca se representa de nuevo como una función de densidad de probabilidad:

$$p(M^o, X_k^a | Y_1, \dots, Y_k, y_{k^o}^o, \dots, y_k^o), \quad (7.22)$$

donde $Y_1, \dots, Y_k$ corresponden con las medidas correspondientes a los punto incluidos en el vector de estados X_k^a . Es decir, se busca la distribución conjunta del vector de estados y las coordenadas del punto M^o , dadas las medidas que se tienen hasta ese momento.

El problema propuesto es conocido como la reconstrucción secuencial en procesos de obtención de estructura a partir de movimiento ó SSFM (“Sequential Structure from Motion”). A pesar de orientarse hacia aplicaciones diferentes, existen soluciones en la literatura equivalentes al propuesto en este capítulo, como los trabajos en SLAM visual presentados por [Davison et al., 2007] [Civera et al., 2007]. En ellos, se demuestra que la distribución (7.22) no puede ser fácilmente aproximada como una función gaussiano y por tanto el filtro de Kalman no es suficiente para resolver el proceso de inferencia, análogo al utilizado cuando se busca la pose $p(X_k^a | Y_1, \dots, Y_k)$.

Para solucionar la no linealidad, resultante de la distribución (7.10), en el trabajo realizado en [Davison et al., 2007], el punto M^o es sustituido por la siguiente parametrización alternativa:

$$M^o = \lambda_k D_k + L_k, \quad (7.23)$$

donde el parámetro λ_k representa la posición o profundidad del rayo que conecta el origen de coordenadas de la cámara O_C con el punto M^o transformado según la pose en el instante k (ver figura 7.7). El vector de tres coordenadas D_k representa el vector unitario, director del rayo y L_k su desplazamiento, siendo ambos referidos con respecto al origen de coordenadas del robot.

Ambos vectores D_k y L_k se denominan observables ya que son obtenidos directamente a partir de los parámetros de la cámara y la pose actual del robot como se muestra a continuación:

$$D_k = g_D(X_k, P, y_k^o) = \frac{t^{-1}(X_k, K^{-1}R_c^T y_k^o)}{\|X_k, K^{-1}R_c^T y_k^o\|} \quad (7.24)$$

$$L_k = g_L(X_k, P) = -t^{-1}(X_k, R^T T_c), \quad (7.25)$$

donde K_c , R_c y T_c corresponden a los parámetros intrínsecos y extrínsecos de la cámara. La función t^{-1} es la transformación inversa de un punto desde el origen de coordenadas del mundo O_W a uno en el origen de coordenadas del robot. Tanto D_k como L_k son observables por la cámara directamente, ya que todos los términos de los que depende, son conocidos. Ambos D_k y L_k forman una recta en el espacio tridimensional que es recorrida por el parámetro de profundidad λ_k , el cual, al contrario que D_k y L_k no es observable mediante una única medida y requiere un conjunto de ellas en diferentes poses del robot,

Puesto que D_k es un vector unitario con dos grados de libertad, admite una parametrización que consiste en dos ángulos ϕ_1 y ϕ_2 :

$$D_k(\phi_k^1, \phi_k^2) = \begin{pmatrix} \cos(\phi_1)\sin(\phi_2) \\ \sin(\phi_1)\sin(\phi_2) \\ \cos(\phi_2) \end{pmatrix}, \quad (7.26)$$

donde ϕ_k^1 y ϕ_k^2 se calculan a partir de g_D como se expone a continuación:

$$\begin{aligned} \phi_k^1 &= \text{atan}(g_D(X_k, P, y_k^o)_2 / g_D(X_k, P, y_k^o)_1) \\ \phi_k^2 &= \text{acos}(g_D(X_k, P, y_k^o)_3) \end{aligned} \quad (7.27)$$

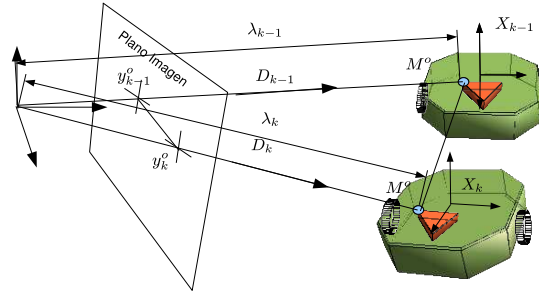


Figura 7.7: Parametrización de profundidad, en dos instantes de tiempo, de un punto candidato a incorporarse al modelo tridimensional.

El punto M^o admite una parametrización diferente en función de la pose del robot, ya que de ésta dependen los valores de D_k y L_k . Sin embargo, el punto M^o calculado mediante la expresión (7.23) es siempre el mismo, puesto que sus coordenadas no varían en un objeto rígido. Se plantea parametrizar M^o en el primer instante de tiempo en el que es visible $k = k^o$, conservando los valores D_{k^o} , L_{k^o} y λ_{k^o} , como incógnitas.

$$M^o = \lambda D_{k^o} + L_{k^o} \quad (7.28)$$

La idea básica del algoritmo es obtener el valor de λ , utilizando para ello el conjunto de medidas $y_{k^o}^o, \dots, y_k^o$. Se define el vector X^M que contiene un total de seis incógnitas que representan la nueva parametrización del punto, utilizando en este caso la descripción de D_{k^o} en sus dos ángulos:

$$X^M = (\lambda \quad \phi_{k^o}^1 \quad \phi_{k^o}^2 \quad L_{k^o}^T)^T \quad (7.29)$$

La nueva distribución de probabilidad “a posteriori” buscada, y que sustituye a (7.10) será:

$$p(X^M, X_k^a | Y_1, \dots, Y_k, y_{k^o}^0, \dots, y_k^o), \quad (7.30)$$

donde cabe destacar que a pesar de que el vector X^M posee tres dimensiones más que la parametrización usual de M^o , tan sólo uno de sus parámetros, es decir λ , no es observable directamente mediante las medidas.

La solución de la distribución conjunta (7.30) requiere ciertas modificaciones en el método “online” convencional, que se enumeran a continuación:

- En primer lugar, el proceso gaussiano equivalente del vector X^M , nombrado como $\mathbf{X}^M$ es obtenido. En k^o la distribución gaussiana para los vectores D_{k^o} y L_{k^o} se obtiene propagando las propiedades estadísticas de X_{k^o} e $y_{k^o}^o$ a través de las funciones no lineales (7.24) y (7.25). Las propiedades de X_{k^o} corresponden a la distribución $p(X_{k^o} | Y_1, \dots, Y_{k^o})$, obtenida por el algoritmo “online” convencional. Las propiedades estadísticas de las medidas del nuevo punto son idénticas a las utilizadas para Y_k .
- El parámetro λ se inicializa suponiendo que el robot carece de dimensión en altura (ver figura 7.8). Su valor de covarianza se fija a un valor de diseño σ_λ^2 , que debe ser lo suficientemente grande como para absorber la incertidumbre que se tiene de λ .

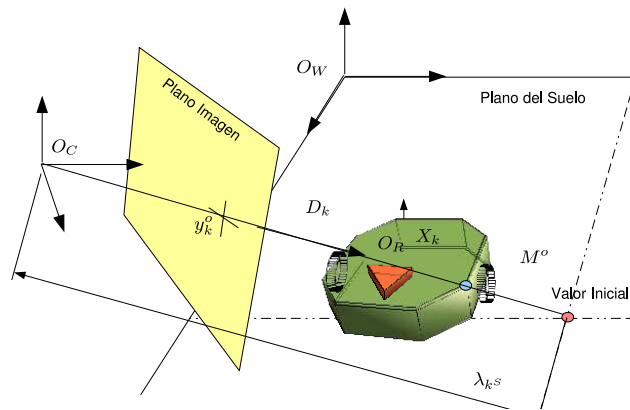


Figura 7.8: Inicialización del parámetro λ suponiendo que el punto está incluido inicialmente en el suelo.

- El vector X^M , incluido en la distribución conjunta con X_k^a , se identificará no sólo por su vector media $\hat{X}^M$ y covarianza Σ^M , sino también por la correlación existente con X_k^a . Ambos vectores forman por tanto un proceso aleatorio gaussiano conjunto representado con el nombre $\mathbf{X}_k^{a,M}$ y cuya función de densidad de probabilidad es (7.22).
- La inferencia Bayesiana utilizada para resolver la distribución

$$p(X_k^{a,M} | Y_1, \dots, Y_k, y_{k^o}^o, \dots, y_k^o),$$

requiere de la definición de un modelo de transición de estados para el vector $X_k^{a,M}$ que modele el movimiento en instantes consecutivos de tiempo. Se define el siguiente modelo extendido:

$$\begin{aligned} X_k^{a,M} &= f^{a,M}(X_{k-1}^{a,M}, U_k) \\ f^{a,M}(X_{k-1}^{a,M}, U_k) &= \begin{pmatrix} f^M(X_{k-1}^M) \\ f^a(X_{k-1}^a, U_k) \end{pmatrix}, \end{aligned} \quad (7.31)$$

donde la función $X_k^M = f^M(X_{k-1}^M)$ es:

$$f^M(X_{k-1}^M) = \begin{pmatrix} \lambda_{k-1} + W_k^\lambda \\ \phi_{k-1}^1 \\ \phi_{k-1}^2 \\ L_{k-1} \end{pmatrix}$$

El índice temporal, añadido al vector X^M , indica que su valor cambia con el tiempo durante el proceso de estimación. El parámetro λ , incluido en dicho vector se modela estadísticamente mediante un proceso aditivo gaussiano de media nula $\mathbf{W}_k^\lambda = N(0, \sigma_{W^\lambda})$. Dicho proceso, incorrelado con el resto de procesos aleatorios, permite que el parámetro λ se ajuste a un valor real mediante los sucesivos procesos de corrección. La varianza de dicho proceso, definida como σ_{W^λ} , controlará con su magnitud las propiedades de convergencia del algoritmo.

- La función de observación h^a se extiende de igual modo en $h^{a,M}$ para incluir los nuevos parámetros X_k^M :

$$\begin{aligned} \begin{pmatrix} y_k^o \\ Y_k \end{pmatrix} &= h^{a,M}(X_k^M, X_k^a) \\ h^{a,M}(X_k^M, X_k^a) &= \begin{pmatrix} h^M(X_k^a, X_k^M) \\ h^a(X_k^a) \end{pmatrix}, \end{aligned} \quad (7.32)$$

$$h^M(X_k^a, X_k^M) = h(t(X_k, \lambda_k D(\phi_k^1, \phi_k^2) + L_k, P)) + v_k^o \quad (7.33)$$

donde v_k^o es un proceso gaussiano de media nula representado por $\mathbf{v}_k^o = N(0, \Sigma_v)$. Este ruido aditivo en las medidas de los puntos nuevos, es al igual que en las componentes del vector Y_k supuestamente incorrelado con el resto de procesos.

- El algoritmo implementa, mediante inferencia Bayesiana, una solución iterativa que calcula el valor de λ que da con la solución para el punto nuevo M^o . Como todo algoritmo de optimización, existe una condición de parada en la que el valor del parámetro a estimar se detiene en un valor concreto. En este caso se define la condición de parada mediante la observación de la covarianza estimada para λ_k , extraída en cada iteración de la matriz $\Sigma_k^{M,a}$. Su valor decrece desde un valor inicial grande, dado al comienzo del algoritmo, hasta un valor cercano a $\sigma_{W\lambda}$.
- Una vez que se considera que el punto obtenido M^o ha alcanzado su valor estimado se propone un método que permite su integración en el vector de estados X_k^a en su parametrización natural (coordenadas tridimensionales). La integración de M^o en el vector de estados requiere encontrar el valor medio de M^o y los términos de covarianza.

El nuevo vector, nombrado temporalmente como X_k^{a+1} se forma mediante la concatenación de X_k^a con el vector M^o :

$$X_k^{a+1} = \begin{pmatrix} X_k^a \\ M^o \end{pmatrix} = f_d(X_k^a, X^M), \quad (7.34)$$

donde f_d corresponde con la función que relaciona X_k^{a+1} con $X_k^{a,M}$:

$$f_d(X_k^a, X^{M^o}) = \begin{pmatrix} X_k^a \\ \lambda_k D_k(\phi_k^1, \phi_k^2) + L_k \end{pmatrix}. \quad (7.35)$$

Mediante la aproximación de primer orden de f_d , se obtienen las matrices de covarianza Σ_k^{a+1} y la media $\hat{X}_k^{a+1}$, las cuales dependen de la distribución gaussiana calculada para el proceso $\mathbf{X}_k^{a,M}$.

El nuevo vector aumentado, que contiene el punto M^o adicional utilizado se renombra para su uso en la siguiente iteración:

$$X_k^a = (X_k, M^1, \dots, M^N, M^{N+1} = M^o) \quad (7.36)$$

En el algoritmo 9 se realiza una breve descripción de la propuesta.

Algoritmo 9 Inicialización de nuevos puntos de la estructura 3D del robot

```

1:  $k - 1 = k^o$ ,  $\mathbf{X}_{k-1}^a$ ,  $N_{it}^o = 0$ 
Require: Medida inicial  $y^o$  del nuevo punto.
2: Inicializar  $D_{k^o}$ ,  $L_{k^o}$  y  $\lambda$  para crear el proceso conjunto  $\mathbf{X}_{k-1}^{a,M}$ .
3: while  $\sigma_k^\lambda < \sigma_{th}$  &  $N_{it} < N_{max}$  do
4:   Obtener  $Y_k$  y  $y_k^o$ .
5:   Usar el Filtro de Kalman para la estimación de  $X_k^{a,M}$  a partir de  $X_{k-1}^{a,M}$ ,  $h^{a,M}$  y  $f^{a,M}$ .
6:   Obtención de la distribución a posteriori  $\mathbf{X}_k^{a,M}$ .
7:    $\sigma_k^\lambda = \sqrt{\Sigma_k^{a,M}(1,1)}$ 
8:    $N_{it}^o = N_{it}^o + 1$ 
9: end while
10: if  $\sigma_k^\lambda < \sigma_{th}$  then
11:   Integrar  $M^o$  en el vector de estados  $X_k^a$  mediante el uso de la función  $f_d$ .
12: end if

```

El algoritmo planteado para un sólo punto M^o es fácilmente extensible a múltiples puntos mediante la búsqueda de la distribución:

$$p(M^{o1}, \dots, M^{or}, X_k^a | Y_{1:k}, y_{k^{o1}:k}^{o1}, \dots, y_{k^{or}:k}^{or} : k) \quad (7.37)$$

En dicho caso, cada nuevo punto, a pesar de estar incluido de manera conjunta en la función de densidad de probabilidad (7.37), su estimación se declara finalizada mediante la inspección del valor de la varianza σ_δ asociada a cada uno por separado.

7.8. Inicialización de la pose o problema del secuestro

El problema del robot secuestrado tiene lugar cuando el sistema pierde completamente la pose del robot y el sistema secuencial de obtención de la pose deja de ser aplicable. A diferencia de los métodos de inicialización comentados en el capítulo 5, cuando el problema del secuestro ocurre se dispone de un modelo tridimensional y de apariencia del robot, aunque no su pose actual.

En el sistema propuesto en esta tesis, el problema del secuestro se detecta en las siguientes dos circunstancias:

- La detección de correspondencias utilizada en el método “online” no obtiene candidatos compatibles en el plano imagen y por tanto se obtiene un vector muy pobre de medida Y_k .
- El sistema de detección obtiene un vector Y_k con suficiente número de correspondencias, de las cuales un número insuficiente sobrevive a la criba realizada por la capa robusta del algoritmo.

Si alguna de las condiciones anteriores se da lugar en un intervalo de tiempo considerable, el robot se considera perdido o “secuestrado”. Existen muchos motivos para que se

produzca una pérdida del robot, como puede ser la oclusión por otros objetos de la escena o una degeneración en el algoritmo de seguimiento. Es ésta última situación la que se pretende resolver con el método propuesto en este apartado.

La solución propuesta en este apartado se basa en utilizar el método de detección “wide-baseline”, propuesto en el capítulo 4 como base para encontrar una correspondencia global entre los puntos del modelo tridimensional del robot y el conjunto de imágenes tomadas por las cámaras.

Cada punto M^i de la estructura del robot, se halla definido mediante las matrices W_t^i y $\hat{d}^i$ que forman la base PCA basada en descriptores SIFT. Dicha base se obtiene durante la inicialización, pudiendo ser refinada secuencialmente, mediante la inclusión de nuevos descriptores en la misma. Mediante al algoritmo 4, descrito en el capítulo 4, se obtiene la correspondencia entre la base PCA de cada punto del modelo tridimensional y los descriptores SIFT encontrados en las imágenes tomadas por el conjunto de cámaras.

Una vez que el proceso de detección ha concluido y se tienen un conjunto de correspondencias suficientes, se obtiene el vector Y_k de medida. A partir de Y_k el objetivo se centra en el cálculo de la pose del robot X_k , desconocida debido al problema del secuestro. El problema que se requiere resolver es por tanto el problema mPnP resuelto en el capítulo 6, con la salvedad de que existen numerosos “outliers” en la medida.

Se propone una solución clásica, implementada frecuentemente en la literatura, en la que se utiliza RANSAC para descartar “outliers” en el vector de medidas obtenido y obtener la pose consensuada por la mayoría de medidas del robot.

El algoritmo a implementar es exacto al utilizado en la capa robusta introducida en el seno del algoritmo de inferencia secuencial y que ha sido descrito en el presente capítulo. Para ello se utiliza la versión del algoritmo mPnP en sus casos de mínimo número de correspondencias (4 en el caso tridimensional y 2 en el caso de movimiento en un plano).

Para ofrecer una visión práctica de la solución planteada, se nombrará como $X_k = f_{PnP}(X_{k-1}^a, Y_k, P)$ a la función utilizada para resolver la pose a partir de los puntos del modelo $M^1, \dots, M^N$, incluido en X_{k-1}^a y las medidas Y_k del conjunto de cámaras.

El algoritmo 10 detalla el proceso de obtención de pose en el problema del secuestro.

Algoritmo 10 El problema del robot secuestrado**Require:** Establecer la probabilidad de confianza necesaria para RANSAC $p = 0,99$

- 1: Obtener los descriptores SIFT de la imagen actual $d_I^1, \dots, d_I^{N_I}$.
- 2: Establecer la correspondencia con la base PCA utilizada como modelo $W_t^1, \dots, W_t^N$ y $\hat{d}^1, \dots, \hat{d}^N$.
- 3: Componer el vector Y_k a partir de la posición detectada de los descriptores.
- 4: Utilizar la función f_{PnP} como función generadora del modelo para un número mínimo de correspondencias.
- 5: Utilizar distancias euclídeas como función distancia entre medidas $y_k^{n,i}$ y la proyección de las mismas usando la pose.
- 6: Algoritmo ransac definido en 8
- 7: **if** $N_{in} > N_{in}^{min}$ **then**
- 8: Calcular el nuevo vector $\mathbf{X}_k^a$ mediante la concatenación del valor X_k y su covarianza calculada mediante el método 7 con los puntos $M^1, \dots, M^N$ y la covarianza Σ_{k-1}^a .
- 9: **end if**

La precisión obtenida mediante la solución propuesta al problema del secuestro recae en el método mPnP presentado en el capítulo 6 y la capacidad para encontrar correspondencias en las imágenes obtenidas mediante el método “wide-baseline”.

7.9. Conclusiones

El presente capítulo aporta una solución secuencial al algoritmo global, la cual se nutre de la inicialización planteada en la tesis y el algoritmo que soluciona el problema mPnP. El algoritmo planteado utiliza un estimador bayesiano, como es el filtro de Kalman para solucionar el problema de la fusión de información de las cámaras y la odometría del robot. Las principales aportaciones de este capítulo se resumen a continuación:

- Se ha aplicado un enfoque de inferencia Bayesiana al problema del cálculo secuencial de la pose del robot, que permite realizar la fusión de la información involucrada, teniendo en cuenta la incertidumbre en cada fuente de información. A su vez, el método secuencial permite facilitar los métodos de obtención de correspondencias entre el modelo tridimensional y el proceso de detección.
- Utilizando los resultados obtenidos en el capítulo 6 se ha propuesto un algoritmo que decrementa en un orden de magnitud la complejidad algorítmica del filtro, utilizando para ello una actualización recursiva de la factorización Cholesky de la matriz de covarianza del vector de estados.
- Se ha propuesto un proceso que obtiene las coordenadas tridimensionales de puntos de la estructura del robot que no fueron considerados en la inicialización. Este algoritmo permite que el conocimiento que se tiene del objeto del que se realiza la localización se incremente a medida que éste muestra su geometría al conjunto de cámaras. La solución propuesta se integra directamente en el estimador bayesiano, lo que permite obtener, además de nuevos puntos en el vector de estados, su incertidumbre y correlación con el resto de puntos del modelo y la pose.

- Mediante la solución al problema mPnP del capítulo 6, el algoritmo de obtención de correspondencias “wide-baseline” se ha propuesto una solución al problema del secuestro.

Capítulo 8

Resultados y validación del sistema propuesto

8.1. Introducción

En este capítulo se presentan los resultados obtenidos en un entorno real de la propuesta de localización en “espacios inteligentes” que ha sido presentada en esta tesis. Se propone realizar experimentos en dos configuraciones de “espacio inteligente”. La primera configuración mostrará los resultados del proceso completo de localización mediante el uso de una cámara. La segunda configuración se centra en el uso de múltiples cámaras.

El capítulo se estructura del siguiente modo: en primer lugar se presenta la tecnología utilizada en los sensores, la plataforma robótica y las comunicaciones que forman una implementación real del “espacio inteligente”. Descrita la tecnología utilizada, se mostrarán los experimentos y sus resultados obtenidos en condiciones reales para las dos configuraciones de una y múltiples cámaras.

8.2. Elementos que componen el “espacio inteligente”

La implementación de “espacio inteligente” que se muestra en este capítulo se compone de un conjunto de cámaras de adquisición sincronizada, robots móviles y una infraestructura de comunicaciones y procesamiento distribuido.

8.2.1. Descripción de las cámaras

Todas las cámaras utilizadas en los experimentos comparten básicamente las mismas características de sensor, y tecnología de adquisición y sincronización de captura. Se trata de cámaras con tecnología de adquisición y control Firewire 1394, utilizando adicionalmente una señal de captura o “trigger” como elemento de sincronización externo. Cada cámara posee un sensor CCD color de 640×480 píxeles de resolución y una velocidad de adquisición de 25 imágenes por segundo. En el experimento de múltiples cámaras se utilizan cámaras “Marlin F-033B/C” [Allied Vision Technologies, 2008], las cuales tienen un tamaño de sensor de tipo 1/2” CCD (8mm de diagonal) con una óptica de 6,5mm. El experimento mediante una cámara utiliza el modelo “DFK 21BF04” [Imaging Source, 2008], con un sensor de tipo 1/4” CCD y una óptica de 4,5mm. Desde el punto de vista de la amplitud de campo, FOV (“Field of View”), ambas cámaras son equivalentes ($\approx 45^\circ$ de FOV).

Calibración de un conjunto de cámaras.

La calibración del conjunto de cámaras consiste en la obtención de los parámetros intrínsecos y extrínsecos de cada una de ellas, fijando un origen del mundo común, en el cual, el plano $x^3 = 0$ coincide con el suelo del “espacio inteligente”. Existen numerosas técnicas que solucionan el problema de la calibración de múltiples cámaras, utilizando para ello patrones métricos y su detección en el plano imagen.

En la presente tesis se ha utilizado el método propuesto por [Heikkila and Silven, 1997], especializado para patrones planos en [Zhang, 1999]. Ambos métodos son eficientemente



Figura 8.1: Cámaras utilizadas en los experimentos

implementados por [Bouguet, 2008] en una herramienta de calibración para el software Matlab y disponible para su libre uso.

La calibración del conjunto de cámaras distingue dos fases: en la primera, independiente por cada cámara, se obtienen los parámetros intrínsecos de la cámara, utilizando para ello un plano de calibración, el cual es observado desde diferentes puntos de vista. Los parámetros intrínsecos obtenidos incluyen una estimación de la distorsión radial existente en las lentes, la cual es considerada en esta tesis como una fuente de incertidumbre en el proceso de estimación. En la figura 8.2 se muestran capturas del patrón de calibración utilizado.

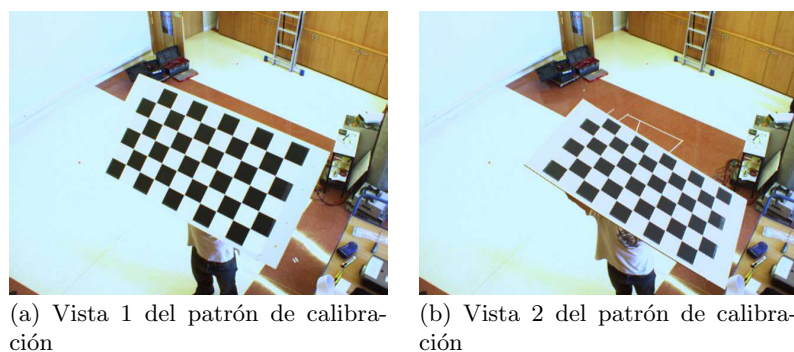


Figura 8.2: Imágenes del patrón de calibración utilizado para obtener los parámetros intrínsecos de una cámara.

La segunda fase utiliza un patrón de calibración que se dispone en el suelo del espacio inteligente, el cual, es observado por el conjunto de cámaras. Mediante dicho patrón se obtiene la posición y orientación de cada cámara con respecto al origen del mundo O_W , el

cual es marcado como una de las esquinas del patrón. En la figura 8.3 se muestra el patrón de calibración de parámetros extrínsecos, visto por cuatro cámaras.

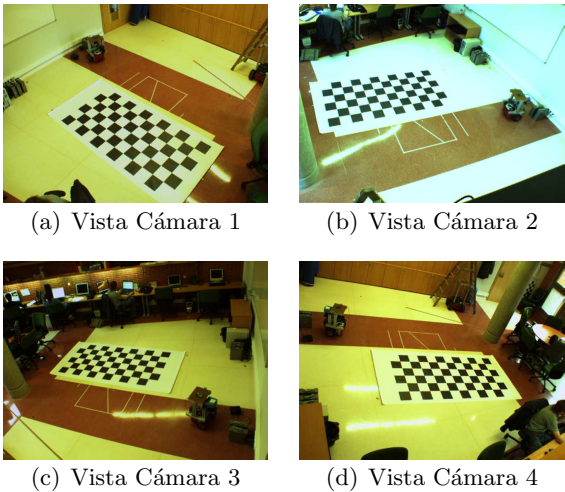


Figura 8.3: Imágenes del patrón de calibración utilizado para calibrar los parámetros extrínsecos

8.2.2. Plataforma robótica

La plataforma robótica utilizada debe disponer de un enlace inalámbrico a través del cual admita consignas de control y ofrezca las lecturas de odometría en tiempo real. Se hará uso de dos plataformas robóticas diferentes en los experimentos realizados. Ambas pertenecen a la empresa “ActivMedia Robotics” y sus características se muestran en la figura 8.4



Características	“Pioneer3-AT”	“Pioneer3-DX”
Dimensiones	50 x 49 x 26	44,5 x 40 x 24,5
Peso	12 Kg	9 Kg
Carga máxima	30 kg	23 kg
V_l máxima	1.2 m/s	1.2 m/s
Batería	12 V	12V
Autonomía	4 - 8 h	18 - 24 h
Tiempo de carga	12 h	12 h
Tracción	4 ruedas motrices, skid-steer	Diferencial
Conexión	WiFi 802.11b o Serie	WiFi 802.11b o Serie

Figura 8.4: Características de las plataformas robóticas utilizadas.

8.2.3. Arquitectura distribuida de adquisición y procesado

La implementación propuesta del “espacio inteligente” requiere la adquisición y procesado de la información de un conjunto de cámaras. La adquisición de éstas debe estar perfectamente sincronizada entre sí y con las lecturas de odometría de los robots controlados por el “espacio inteligente”. Se ha implementado una arquitectura cliente-servidor que permite la adquisición y procesado de las cámaras, repartiendo la carga necesaria, tanto en términos de ancho de banda como en el procesado de las imágenes.

Las cámaras se dividen en conjuntos que son controlados por un nodo de procesamiento que adquiere las imágenes a través del bus de adquisición (IEEE1394). En dicho nodo se realiza el procesamiento de bajo nivel de las imágenes, como puede ser la obtención de marcas naturales de la imagen. Dentro de cada nodo de adquisición y procesado se implementa un servidor TCP/IP que sirve la información obtenida de cada imagen a un cliente. Dicho cliente, el cual implementa el resto del algoritmo, es el encargado de fundir la información proveniente de los servidores con la odometría obtenida del robot, que implementa un servidor TCP/IP a través del enlace inalámbrico. La arquitectura planteada permite otorgar una gran flexibilidad, dado que el cliente de procesado puede estar en un lugar físico diferente a los servidores de adquisición. La sincronización de las cámaras se realiza a través de una línea dedicada que se activa a través de uno de los servidores, el cual recibe la orden de adquisición a través del bus de comunicaciones que lo conecta con el cliente.

8.3. Experimento utilizando una cámara

En esta sección se mostrarán los resultados del proceso completo de localización mediante una cámara. El experimento mostrará los resultados del proceso de inicialización presentado en el capítulo 5 y el resultado de aplicar el algoritmo de localización secuencial del robot, presentado en el capítulo 7. El modelo tridimensional, obtenido durante la inicialización será aumentado mediante la incorporación de nuevos puntos, como ha sido presentado en la sección 7.7.

Las condiciones del experimento son las siguientes:

- El escenario propuesto para el experimento utiliza el modelo de cámara “DFK-21BF04” y el robot “Pioneer3-DX”. La cámara se encuentra dispuesta con respecto al origen de coordenadas del mundo como se muestra en la figura 8.5. Los parámetros de calibración de la misma se muestran a continuación:

$$P = \begin{cases} \alpha_c = 15,7918^\circ & \beta_c = -104,8724^\circ & \gamma = 77,2273 \\ T_c = (675,7848 & 572,1395 & 2911,9)^T \\ f_u = 838,0493 & f_v = 838,9801 & u_0 = 363,4370 & v_0 = 233,5077 \end{cases} \quad (8.1)$$

- El robot describe una trayectoria, cuya estimación realizada mediante la odometría se muestra en la figura 8.6. Se utilizarán diversos tramos de dicha trayectoria para evaluar el algoritmo de inicialización.

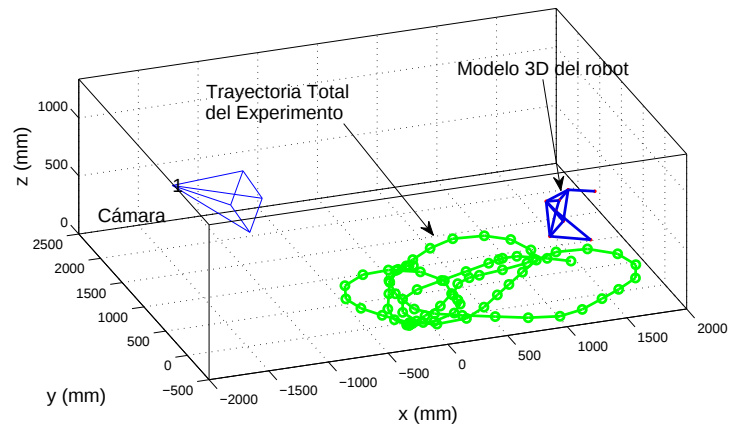


Figura 8.5: Distribución geométrica de la cámara y la trayectoria del robot.

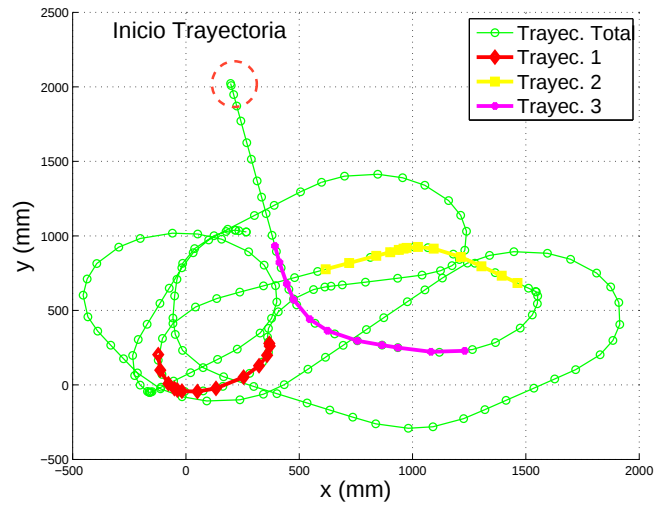


Figura 8.6: Trayectorias de inicialización comparadas con la trayectoria total realizada por el robot.

- En la figura 8.7 se muestra un modelo tridimensional del robot compuesto por $N = 8$ puntos de la estructura tridimensional del mismo, los cuales son fácilmente identificables por el ojo humano en una imagen. Las coordenadas tridimensionales del modelo se obtienen por medición directa en la plataforma real. Dicho modelo será utilizado como medida de “ground-truth” o referencia de la pose. Para ello se identifican manualmente los puntos del modelo de referencia en el plano imagen de las diferentes cámaras. Utilizando el algoritmo mPnP, descrito en el capítulo 6, como medida de la rotación y traslación del robot, se obtiene una estimación de la pose, la cual se utilizará como referencia. La medida de “ground-truth” planteada, aunque afectada por el error de anotación y la calibración de la cámara, permite no obstante disponer de una estimación precisa de la pose real que ocupa el robot y que no contiene error acumulativo.

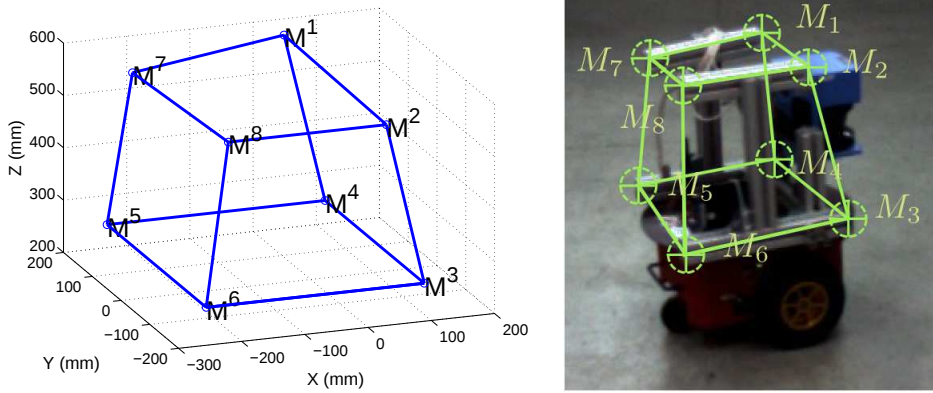


Figura 8.7: Modelo tridimensional utilizado como referencia, obtenido mediante medición directa sobre la plataforma real y su anotación manual en la imagen

- El error esperado en las lecturas de odometría depende de diversos factores, como el tipo de plataforma robótica utilizada, la fricción entre las ruedas y la superficie de desplazamiento y el propio error y cuantificación existente en los sensores utilizados. En el experimento planteado se supone una desviación típica del error en la velocidad lineal $\sigma_{v_l} = 20mm$, y una desviación angular $\sigma_{\Omega} = 4^\circ$, los cuales se ajustan experimentalmente al comportamiento que posee la plataforma robótica.
- El error de medida σ_v^2 , que determina la incertidumbre en la detección por cada coordenada de la imagen se establece en $\sigma_v = 10 \text{ pixels}$.

8.3.1. Experimento de inicialización

Se proponen tres trayectorias de inicialización diferentes, las cuales se muestran en la figura 8.6. En cada una de ellas se implementará el algoritmo de inicialización, utilizando la matriz de covarianzas completa. El modelo tridimensional obtenido en cada uno de los experimentos se muestra en las figuras 8.8.a, 8.8.c y 8.8.e. La reproyección del modelo tridimensional obtenido se muestra en 8.8.b, 8.8.d y 8.8.f.

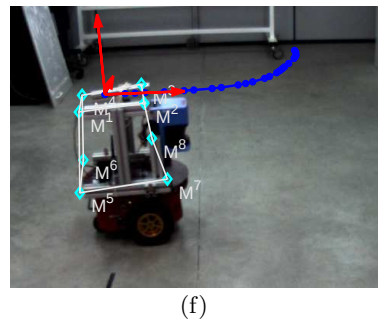
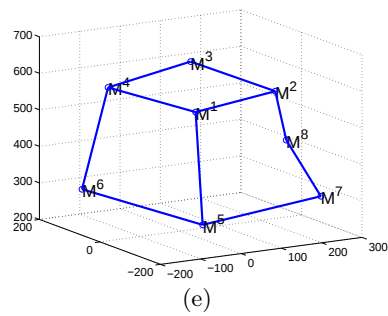
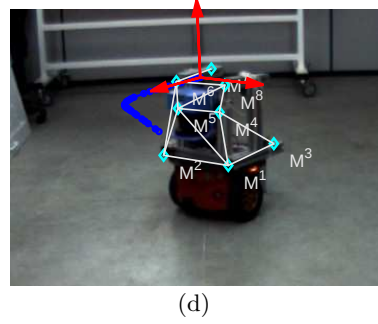
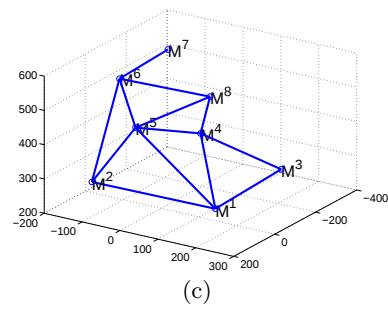
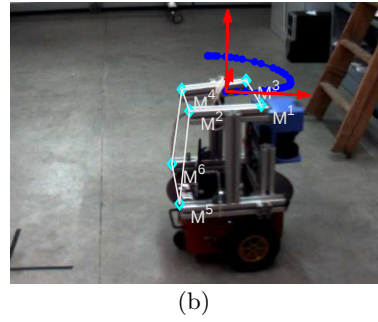
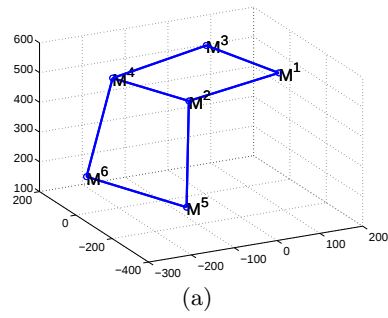


Figura 8.8: Resultados del algoritmo de inicialización mediante una cámara. En cada fila se muestran los resultados obtenidos, utilizando una de las tres trayectorias de inicialización.

8.3.2. Experimento de localización secuencial

El experimento de localización secuencial parte del modelo tridimensional del robot, obtenido mediante la segunda trayectoria de inicialización, mostrada anteriormente en la figura 8.6 y que produce como resultado el modelo tridimensional del robot, mostrado en la figura 8.8.c.

En la figura 8.9 se muestra la posición estimada por ambos métodos de posicionamiento secuencial, diferenciados por su complejidad cúbica y cuadrática, comentados en el capítulo 7. La posición del robot, obtenida por ambos métodos se compara con la estimación de odometría y la medida de “ground truth” obtenida mediante anotación en imágenes. Como se observa, ambos métodos obtienen similares resultados, muy cercanos a la medida de referencia, y claramente diferenciados de la estimación mediante odometría.

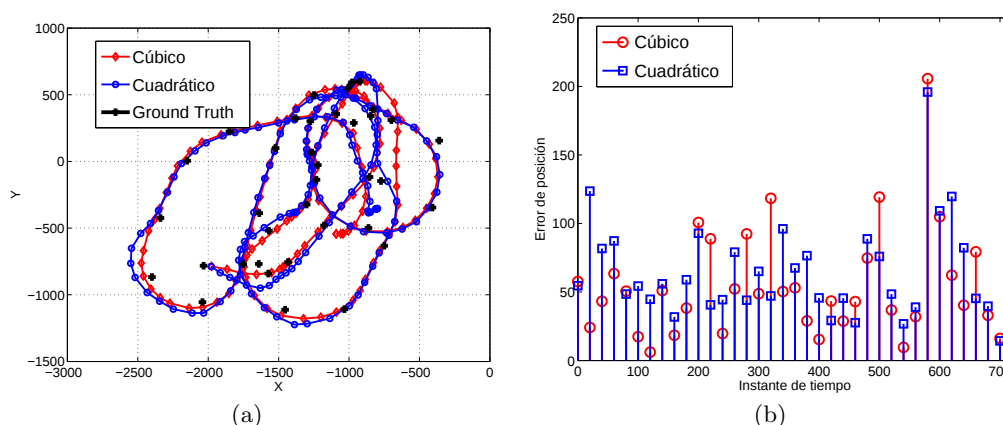


Figura 8.9: Comparación de las trayectorias obtenidas mediante las dos alternativas de método secuencial, comparadas con la medida de referencia.

Durante el experimento, el modelo tridimensional del robot es aumentado mediante el método comentado en el capítulo 7, obteniendo como resultado el modelo tridimensional que se muestra en la figura 8.10, para ambos métodos. En las figuras 8.11 se muestran algunos instantes de tiempo de la trayectoria, con el objetivo de mostrar la precisión en la reproyección de la imagen.

8.3.3. Experimento en una secuencia que incluye oclusiones

A continuación se propone un experimento utilizando una sola cámara y el mismo robot en una configuración similar a la presentada en el punto anterior, salvo por el hecho de que la secuencia propuesta presenta una longitud mayor y en la escena se producen oclusiones. En la figura 8.12.a se muestra la disposición geométrica de la cámara, el robot y la trayectoria total realizada por el mismo. En la figura 8.12.b se muestra un modelo tridimensional del robot, resultado del proceso de inicialización en la zona de la trayectoria remarcada en la figura 8.12.a.

El resultado de localización, en comparación con la medida de referencia se muestra

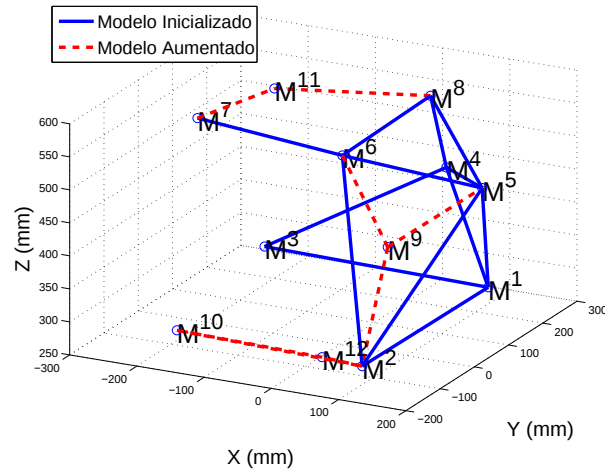


Figura 8.10: Modelo 3D aumentado durante el proceso secuencial

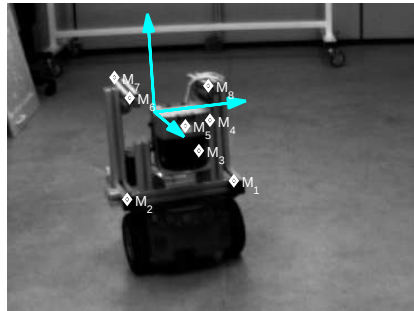
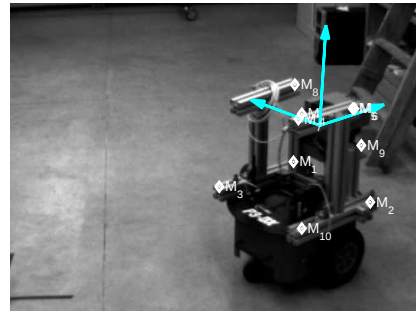
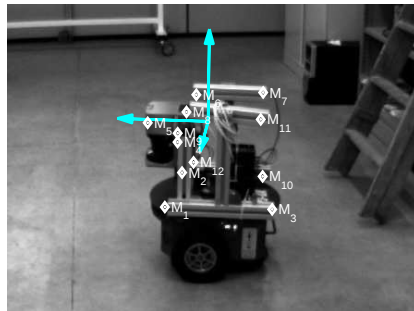
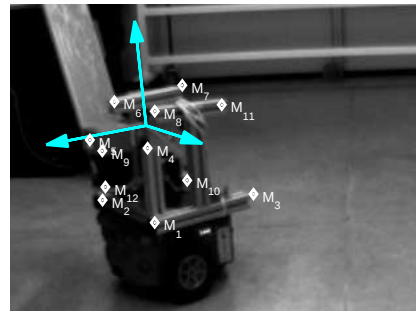
(a) $k = 250$ (b) $k = 308$ (c) $k = 350$ (d) $k = 460$

Figura 8.11: Reproyección del modelo tridimensional durante diversos frames de la trayectoria de posicionamiento.

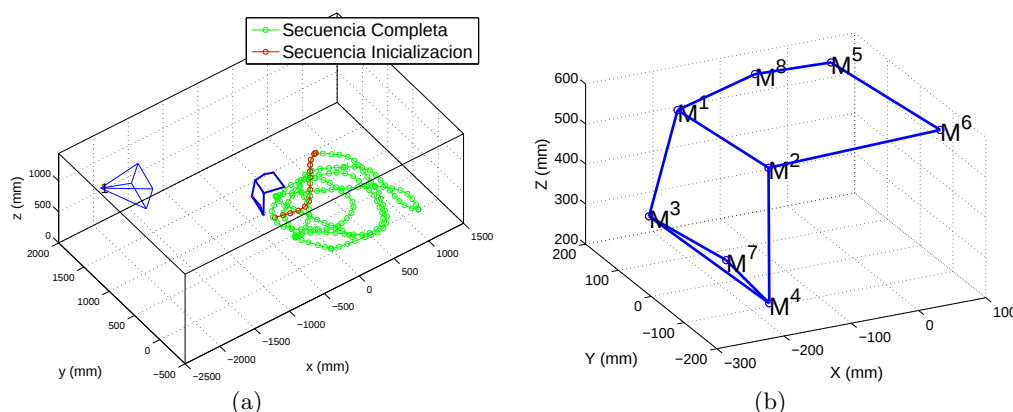


Figura 8.12: Disposición geométrica de la cámara en el experimento y modelo tridimensional obtenido

en dos figuras 8.13.a y 8.13.b, en las que se muestran respectivamente la comparación de la trayectoria estimada frente a puntos de referencia y el error en dichos instantes de referencia. En la figura 8.14 se muestran varios instantes de tiempo en los que se producen oclusiones en la escena.

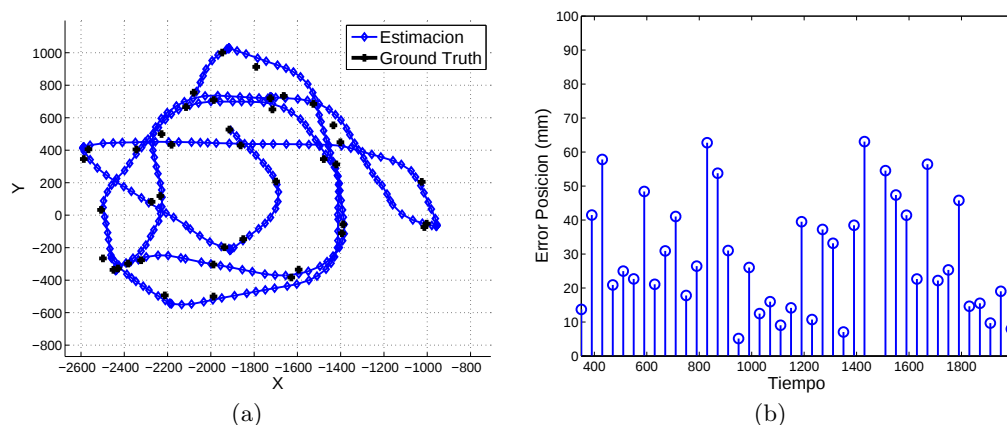


Figura 8.13: Localización del robot en una escena donde existen oclusiones

Como se desprende de las figuras 8.13.a y 8.13.b, la precisión obtenida en la localización no presenta error acumulativo y su error con respecto a la medida de referencia utilizada permanece relativamente constante. Las oclusiones mostradas en la figura 8.14 muestran que el sistema no se ve afectado significativamente por ellas y aunque en los instantes de oclusión también se carece de medida de referencia, los instantes adyacentes poseen un error de localización comparable al resto de la secuencia.

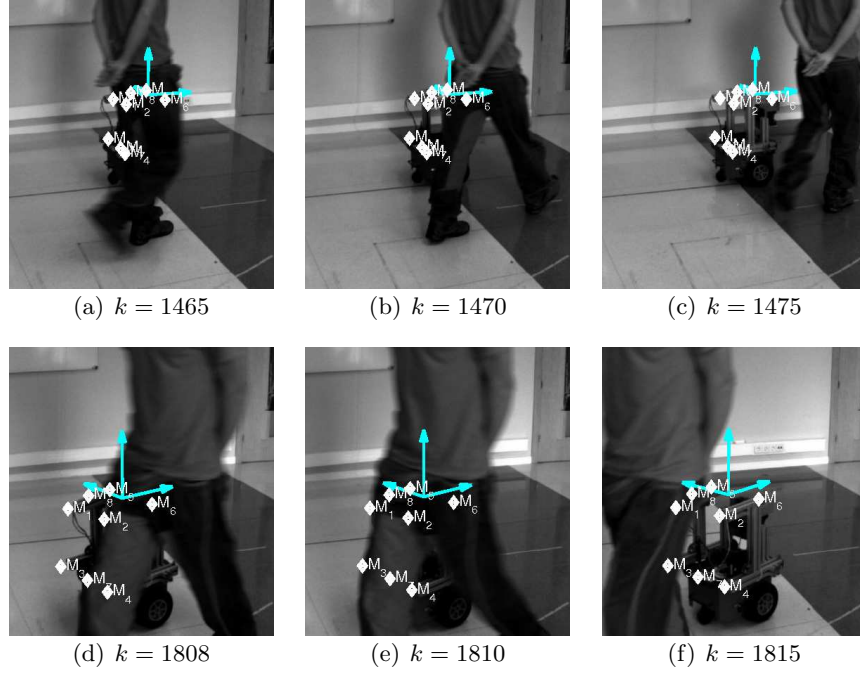


Figura 8.14: Ejemplo de oclusiones y su influencia en el posicionamiento

8.4. Experimento utilizando un entorno de múltiples cámaras

El experimento en el entorno real planteado para múltiples cámaras hace uso de cuatro cámaras dispuestas como se muestra en la figura 8.15.a. Para realizar este experimento se utilizan las cámaras “Marlin F33C” y el modelo de robot “Pioneer3-AT”.

La trayectoria de inicialización es mostrada en la figura 8.15.a junto con la posición tridimensional del robot y las cámaras. Los parámetros de calibración de las cámaras se muestran a continuación:

$$P^1 = \begin{cases} \alpha_c^1 = -36,8264^\circ & \beta_c^1 = -33,3948^\circ & \gamma_c^1 = -140,1743^\circ \\ T_c^1 = (396,7794 & 1180,8 & 3643,1)^T \\ f_u^1 = 631,3688 & f_v^1 = 637,0136 & u_0^1 = 320,2614 & v_0^1 = 228,2005 \end{cases} \quad (8.2)$$

$$P^2 = \begin{cases} \alpha_c^2 = 56,4173^\circ & \beta_c^2 = 41,1233^\circ & \gamma_c^2 = -145,7345^\circ \\ T_c^2 = (-1999,4 & -54,2618 & 5127,6)^T \\ f_u^2 = 642,7650 & f_v^2 = 643,7740 & u_0^2 = 322,9353 & v_0^2 = 211,0803 \end{cases} \quad (8.3)$$

$$P^3 = \begin{cases} \alpha_c^3 = 121,9742^\circ & \beta_c^3 = 46,3774^\circ & \gamma_c^3 = 146,5448^\circ \\ T_c^3 = (-1320,2 & -278,2942 & 6744,3)^T \\ f_u^3 = 636,0547 & f_v^3 = 636,5856 & u_0^3 = 324,5558 & v_0^3 = 237,8806 \end{cases} \quad (8.4)$$

$$P^4 = \begin{cases} \alpha_c^4 = -114,6361^\circ & \beta_c^4 = -45,4540^\circ & \gamma_c^4 = 143,6925^\circ \\ T_c^4 = (1336,9 & 804,3017 & 5442,2)^T \\ f_u^4 = 635,4452 & f_v^4 = 637,3190 & u_0^4 = 330,5558 & v_0^4 = 217,0185 \end{cases} \quad (8.5)$$

En la figura 8.15.b se muestra el modelo tridimensional obtenido, el cual consta de $N = 21$ puntos.

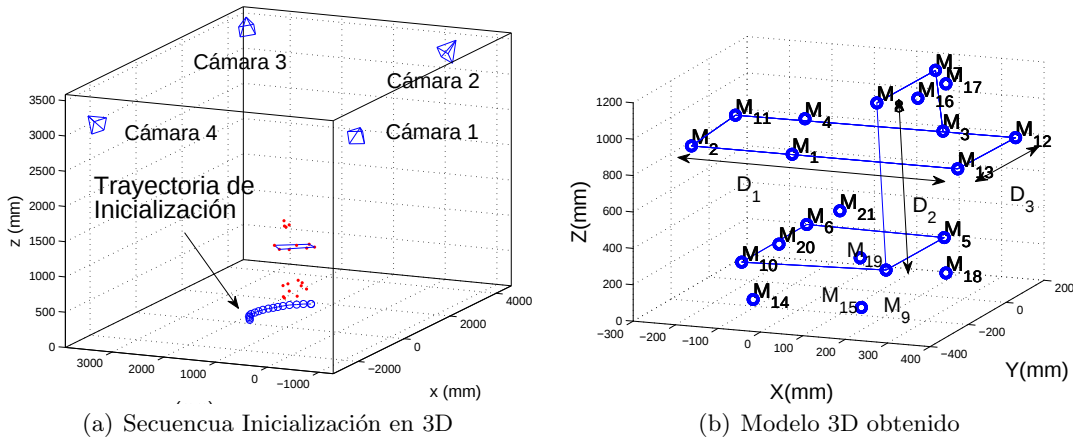
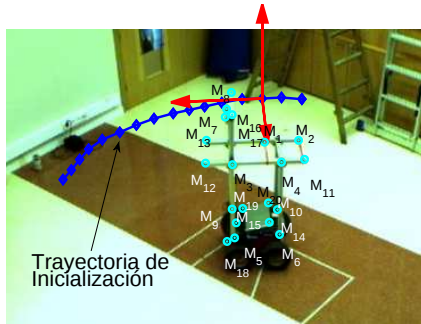


Figura 8.15: Trayectoria y modelo tridimensional resultante de la inicialización mediante múltiples cámaras.

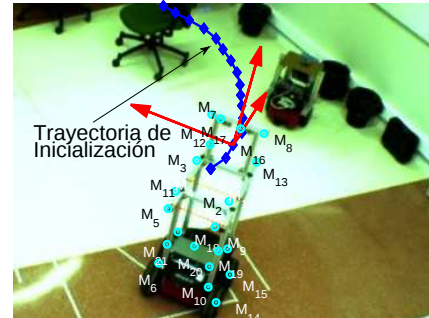
En las figuras 8.16.a, 8.16.b, 8.16.c y 8.16.d se muestra dicho modelo reproyectado en un mismo instante de tiempo en cuatro cámaras. Se muestra a su vez la trayectoria de inicialización proyectada sobre el plano imagen de cada cámara.

En la figura 8.15.b se muestran tres medidas de distancias D_1 , D_2 y D_3 , obtenidas en la plataforma robótica real con el objetivo de crear una medida de precisión o “ground truth”, que permita comparar el algoritmo de localización con un sistema de posicionamiento de referencia. Mediante dichas medidas y un sistema que permite indicar manualmente a un usuario la posición de los puntos M_{12} , M_{13} , M_5 , M_9 , M_2 , se obtiene la rotación y traslación del robot, utilizando para ello el algoritmo $mPnP$, descrito en el capítulo 6. Al igual que la medida de referencia utilizada en el experimento de una sola cámara, el “ground-truth” utilizado depende de los parámetros de calibración de las cámaras y los posibles errores del usuario en acertar la posición exacta de los puntos del robot, permitiendo en todo caso acotar la precisión obtenida por el algoritmo.

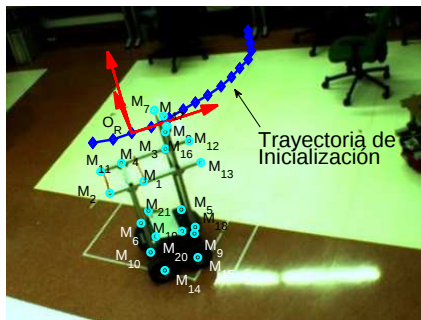
En la figura 8.17, se muestra una comparación entre la medida de “ground truth” y el resultado de aplicar el método online. El robot realiza una trayectoria en la que un humano



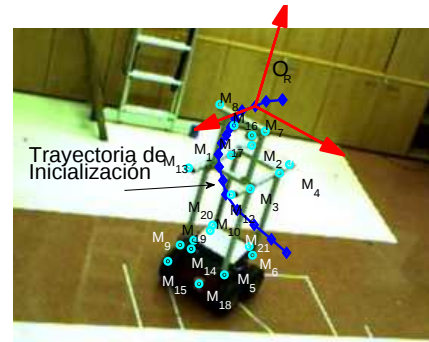
(a) Secuencia de inicialización en la cámara 1



(b) Secuencia de inicialización en la cámara 2



(c) Secuencia de inicialización en la cámara 3



(d) Secuencia de inicialización en la cámara 4

Figura 8.16: Modelo tridimensional reproyectado junto con la trayectoria utilizada por el algoritmo de inicialización mediante múltiples cámaras.

camina alrededor del robot varias veces con el objetivo de generar oclusiones, como se muestra en la figura 8.18 para una secuencia de imágenes. El algoritmo secuencial se ha probado utilizando cuatro cámaras, un conjunto de tres cámaras (1,2,3) y dos cámaras (1,2). El resultado se compara con una medida de referencia, o “ground truth”. En las figuras 8.17.a y en la figura 8.17.b se muestra el error de posición en función del instante de tiempo, con el objetivo de mostrar que las oclusiones producidas por el usuario humano (en los instantes de tiempo indicados en el pie de la figura 8.17), no afectan sustancialmente al resultado de posicionamiento.

En todo el experimento se utiliza el método de complejidad cuadrática, puesto que su precisión en este caso demuestra ser equivalente al caso de complejidad cúbica.

En la figura 8.19, se muestra un experimento donde el robot recorre una trayectoria en la que existen también cambios de iluminación y oclusiones, utilizando para ello la información proveniente de tres cámaras. En la figura 8.19.a se muestra la posición del robot junto con la medida de “ground truth” y en la figura 8.19.b se muestra el tipo de cambio de iluminación producido en el experimento.

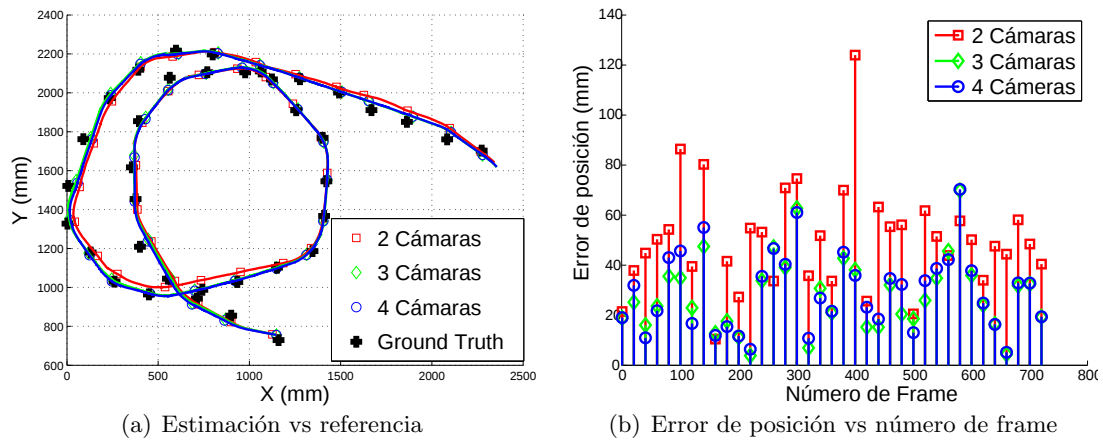


Figura 8.17: Comparación entre el “ground truth” y el resultado de localización mediante múltiples cámaras

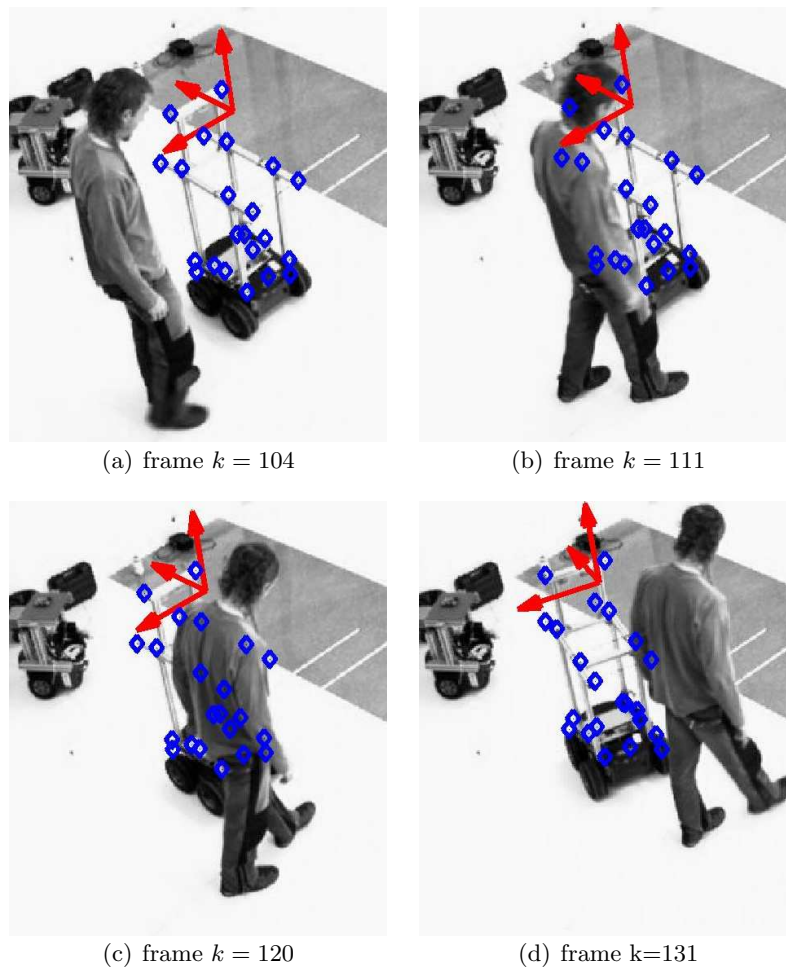


Figura 8.18: Ejemplo del tipo de oclusiones ocasionadas por el usuario humano al rodear al robot.

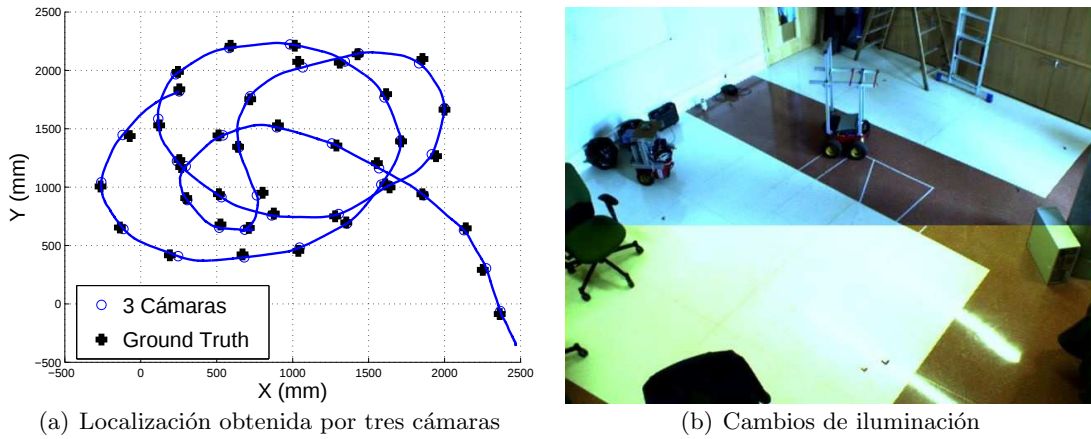


Figura 8.19: Experimento con cambios de iluminación y tres cámaras

8.5. Conclusiones

El presente capítulo presenta una validación experimental del sistema de posicionamiento completo, propuesto como objetivo en esta tesis. Se han presentado dos escenarios, uno centrado en el sistema de posicionamiento de un robot mediante una cámara y otro escenario en el que se dispone de un anillo de hasta cuatro sensores y un robot que se encuentra a una distancia a cada sensor superior al experimento monocular.

Los resultados obtenidos en ambos escenarios demuestran que el sistema de localización propuesto es preciso y estable en secuencias extensas en las que se incluyen cambios de iluminación y oclusiones. Aunque la medida de referencia utilizada está influenciada por los errores de calibración, ésta es útil para demostrar que el sistema no posee un error acumulativo o una tendencia degenerativa en la estimación. Se ha mostrado, en el caso de una cámara, cómo el algoritmo de complejidad cuadrática posee en este caso real una precisión comparable a la versión “convencional” que posee una complejidad cúbica, siendo por tanto el proceso utilizado para el resto de simulaciones incluido el caso de múltiples cámaras.

Como se ha demostrado en este capítulo, (ver figuras 8.6 y 8.15), la trayectoria de inicialización necesaria es de reducido tamaño y por tanto el tiempo que el robot se encuentra sin localizar permite su aplicación en entornos reales. El sistema propuesto constata que un conjunto de $N = 6$ o $N = 8$ puntos, obtenidos mediante métodos de “tracking” de marcas naturales, es suficiente para la inicialización del sistema de localización y un posterior proceso secuencial estable. Se ha mostrado a su vez el resultado de aplicar el método de ampliación del modelo tridimensional del robot integrado en el proceso de localización secuencial, el cual produce un modelo aumentado sin repercutir en la estabilidad del método propuesto. El modelo tridimensional aumentado que se obtiene, demuestra tener una precisión del mismo orden que el obtenido mediante el método de inicialización, a juzgar por la correcta reproyección de dichos puntos en la secuencia de localización.

Aunque en ningún caso se especifican los tiempos de proceso, el algoritmo propuesto admite una implementación en tiempo real, siendo el detector SIFT el principal cuello de

botella del proceso. El filtro, incluida la capa robusta mediante RANSAC, aporta una baja complejidad debido al pequeño número de características utilizadas del robot. En todo caso, los experimentos propuestos demuestran que el filtro con complejidad cuadrática es aplicable a casos reales, sin que ello signifique una pérdida de precisión sustancial. Dicho filtro puede significar un ahorro de recursos en implementaciones mediante hardware de bajo coste o modelos tridimensionales con un gran número de puntos.

Capítulo 9

Conclusiones

9.1. Introducción

En este capítulo se exponen las conclusiones obtenidas en cada uno de las propuestas realizadas en la tesis, así como un resumen de las aportaciones realizadas. El presente capítulo incluye a su vez una disertación sobre las líneas de investigación que se pueden continuar a partir del trabajo realizado en la tesis.

9.2. Conclusiones y aportaciones de la tesis

En la presente tesis se ha presentado un algoritmo completo de localización de robots en “espacios inteligentes” a partir de cámaras fijas en el entorno. El principal objetivo propuesto en la tesis es la definición de un algoritmo de localización que minimice el conocimiento “a priori” que se requiere del robot a localizar. Este enfoque es novedoso en la literatura sobre “Espacios Inteligentes”, ya que permite localización mediante cámaras sin necesidad de recurrir a marcas artificiales dispuestas sobre el robot. A continuación se destacan los objetivos planteados y las aportaciones realizadas.

- El algoritmo de medida en el plano imagen que se ha planteado en esta tesis tiene como objetivo encontrar de manera eficiente la proyección de puntos de la estructura del robot en el conjunto de cámaras existentes. La principal aportación realizada a ese respecto es la integración de diversas técnicas dedicadas a la extracción de marcas naturales (como son los descriptores SIFT y los algoritmos basados en KLT) para su uso en el problema de “tracking” de objetos tridimensionales, donde las aproximaciones de estructura plana, local a cada punto, no son aplicables.
- Previo al proceso de localización, se plantean procesos de inicialización que tienen como objetivo encontrar la pose inicial del robot y un conjunto de puntos de su estructura tridimensional, identificados mediante el proceso de medida en el plano imagen.

El algoritmo de inicialización mediante una sola cámara, posee como aportación fundamental el uso de un enfoque de máxima verosimilitud que permite utilizar la odometría como referencia métrica en la reconstrucción. Se utiliza una técnica de “Bundle Adjustment” con una función de coste adecuada al criterio de verosimilitud, utilizando para ello una matriz de ponderación, la cual permite disminuir el impacto del error de la odometría en la solución. Dicha matriz y sus diversas aproximaciones demuestran diferente influencia en la precisión y complejidad matemática del algoritmo. Se ha propuesto una modificación de la función de coste para incluir una distribución menos sensible a la presencia de “outliers” en la medida y se ha propuesto un algoritmo no iterativo y exacto de inicialización previa a la técnica de “Bundle Adjustment”. A su vez se han demostrado matemáticamente situaciones en las que la trayectoria efectuada por el robot produce una solución degenerada.

La inicialización mediante múltiples cámaras planteada, aporta la posibilidad de obtener la pose y geometría del robot en distribuciones de cámaras alejadas y no solapadas, ya que no requiere correspondencias entre las mismas. Al igual que en el caso mono-

cular, se han propuesto versiones robustas del algoritmo y métodos de inicialización lineales.

- A partir de la información obtenida durante la inicialización, se ha diseñado un método de localización secuencial del robot. Se han propuesto dos algoritmos necesarios para la localización:

Se ha diseñado un algoritmo que soluciona el problema mPnP, que obtiene la pose de un objeto a partir de su modelo tridimensional y la proyección de éste en el plano imagen de múltiples cámaras. El algoritmo planteado para resolver el problema mPnP amplía el trabajo propuesto por [Moreno-Noguer et al., 2007] para su uso en múltiples cámaras. El algoritmo propuesto conserva las excelentes propiedades de precisión y complejidad lineal del algoritmo con el número de correspondencias. Se ha propuesto a su vez una solución para movimientos restringidos a un plano y se ha demostrado el incremento en precisión que aporta, además de permitir soluciones para el problema de los 2 puntos. Como añadido, se ha propuesto un método que permite evaluar y estimar la incertidumbre que existe en la solución de la pose, dada la incertidumbre en las medidas y el modelo tridimensional.

El segundo algoritmo presentado obtiene la pose del robot de manera secuencial. Tiene como aportación la aplicación de un enfoque de inferencia Bayesiana al problema. De este modo se realiza una sinergia coherente entre las fuentes de información disponibles. Utilizando la solución al problema mPnP se ha propuesto un algoritmo que decrementa en un orden de magnitud la complejidad algorítmica del filtro así como un método robusto que elimina los “outliers” del vector de medidas.

- Se ha propuesto un proceso que obtiene las coordenadas tridimensionales de puntos de la estructura del robot que no fueron considerados en la inicialización. Este algoritmo permite que el conocimiento que se tiene del objeto del que se realiza la localización se incremente a medida que éste muestra su geometría al conjunto de cámaras. La solución propuesta se integra directamente en el estimador Bayesiano, lo que permite obtener, además de nuevos puntos en el vector de estados, su incertidumbre y correlación con el resto de puntos del modelo y la pose.
- Mediante la solución al problema mPnP del capítulo 6, el algoritmo de obtención de correspondencias “wide-baseline” se ha propuesto una solución al problema del secuestro.
- Mediante experimentos en entornos reales se ha confirmado la validez de la propuesta como sistema de localización de robots móviles.

9.3. Líneas de investigación futuras derivadas de la tesis

Es interesante destacar algunas de las líneas de investigación futuras que se plantean a partir del trabajo realizado en esta tesis.

- El algoritmo presentado de localización posee como restricción fundamental la suposición de rigidez en el elemento sobre el que se realiza el posicionamiento. Este hecho

impide el uso de una variación del algoritmo para su utilización en el posicionamiento de usuarios, puesto que éstos presentan una estructura deformable. En la actualidad existe una gran actividad investigadora en sistemas de reconstrucción para objetos deformables y por tanto la creación de un sistema con la misma filosofía que la presentada en la presente tesis pero aplicada a dicho tipo de problema es una línea futura de investigación.

- El sistema propuesto depende en gran medida de que el objeto tenga un tamaño suficiente en la imagen de las cámaras, de tal modo que su geometría pueda ser inferida o que su estructura presente suficiente contraste para ofrecer puntos de interés para su reconstrucción. La técnica presentada en esta tesis permite ser combinada con técnicas de reconstrucción basadas en “Visual Hull”, las cuales pueden entregar un volumen acotado donde está localizado el robot. Mediante dichas técnicas se puede tener localización en objetos pequeños o sin textura, al mismo tiempo que se robustece la propuesta de esta tesis.
- En la tesis se ha utilizado el descriptor SIFT, el cual posee una complejidad algorítmica elevada. Existen en la literatura otros tipos de técnicas de detección de marcas naturales y sus descriptores invariantes, como puede ser el descriptor SURF [Bay et al., 2006]. El uso de dichas técnicas permite disminuir el tiempo de cómputo requerido por el sistema de localización.
- En la presente tesis se realizan modelos tridimensionales basados en puntos, como única primitiva geométrica, a partir de la cual realizar el posicionamiento. En los objetos manufacturados es muy común la existencia de un conjunto elevado de líneas rectas, y por tanto su inclusión en el modelado y el vector de estados aportaría precisión y estabilidad.
- Finalmente, la solución planteada ha sido evaluada en estancias reducidas y no se ha considerado el problema de cubrir un espacio grande con múltiples cámaras asociadas en diferentes espacios cerrados. La estabilidad del algoritmo en ese caso y la problemática extra derivada de su implementación es una línea futura de trabajo para la implantación de este tipo de sistemas en edificios.

Bibliografía

- [Ahn and Yu, 2006] Ahn, H. and Yu, W. (2006). Wireless Localization Network for a Ubiquitous Robotic Space: Background and Concept. *Proc. of the 3rd International Conference on Ubiquitous Robots and Ambient Intelligence, Seoul, Korea*, páginas 187–192.
- [Allied Vision Technologies, 2008] Allied Vision Technologies (2008). Marlin f-033 b/c. <http://www.alliedvisiontec.com/avt-products/cameras/marlin/f-033-b-bs-c.html>.
- [Ara et al., 2008] Ara, K., Kanehira, N., Olguin, D. O., Waber, B.Ñ., Kim, T., Mohan, A., Gloor, P., Laubacher, R., Oster, D., Pentland, A. S., and Yano, K. (2008). Sensible organizations: Changing our businesses and work styles through sensor data. *Journal of Information Processing (Information Processing Society of Japan)*, 16:1–12.
- [Armstrong and Zisserman, 1995] Armstrong, M. and Zisserman, A. (1995). Robust object tracking. En *Asian Conference on Computer Vision*, páginas 58–62.
- [Baker and Matthews, 2004] Baker, S. and Matthews, I. (2004). Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, páginas 221–255.
- [Basu et al., 1996] Basu, S., Essa, I., and Pentland, A. (1996). Motion regularization for model-based head tracking. En *International Conference on Pattern Recognition*, Vienna, Austria.
- [Baumberg, 2000] Baumberg, A. (2000). Reliable feature matching across widely separated views. En *Conference on Computer Vision and Pattern Recognition*, páginas 774–781.
- [Bay et al., 2006] Bay, H., Tuytelaars, T., Gool, V., and Surf, L. (2006). Speeded up robust features. *Proceedings of the 9th European Conference on Computer Vision*, 1:404–417.
- [Beis and Lowe, 1997] Beis, J. and Lowe, D. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. En *Conference on Computer Vision and Pattern Recognition*, páginas 1000–1006, Puerto Rico.
- [Black and Yacoob, 1997] Black, M. J. and Yacoob, Y. (1997). Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, páginas 23–48.
- [Bouguet, 2008] Bouguet, J.-Y. (2008). Matlab calibration toolbox. http://www.vision.caltech.edu/bouguetj/calib_doc/.

- [Briggs et al., 2000] Briggs, A., Scharstein, D., Braziunas, D., Dima, C., and Wall, P. (2000). Mobile robot navigation using self-similar landmarks. *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 2.
- [Cascia et al., 2000] Cascia, M., Sclaroff, S., and Athitsos, V. (2000). Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22.
- [Chang and Chen, 2004] Chang, W.-Y. and Chen, C.-S. (2004). Pose Estimation for Multiple Camera Systems. En *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3*, páginas 262–265, Washington, DC, USA. IEEE Computer Society.
- [Chia et al., 2002] Chia, K., Cheok, A., and Prince, S. (2002). Online 6 DOF augmented reality registration from natural features. En *International Symposium on Mixed and Augmented Reality*.
- [Cho et al., 1998] Cho, Y., Lee, W., and Neumann, U. (1998). A multi-ring color fiducial system an intensity-invariant detection method for scalable fiducial-tracking augmented reality. En *International Workshop on Augmented Reality*.
- [Christy and Horaud, 1999] Christy, S. and Horaud, R. (1999). Iterative Pose Computation from Line Correspondences. *Computer Vision and Image Understanding*, 73:137–144.
- [Civera et al., 2007] Civera, J., Davison, A., and Montiel, J. (2007). Inverse Depth to Depth Conversion for Monocular SLAM. *Robotics and Automation, 2007 IEEE International Conference on*, páginas 2778–2783.
- [Coen, 1998] Coen, M. H. (1998). Design principles for intelligent environments. En *Proceedings of the fifteenth national, tenth conference on Artificial intelligence Innovative applications of artificial intelligence*, páginas 547–554, Menlo Park, CA, USA. American Association for Artificial Intelligence.
- [Crowley, 1982] Crowley, J. (1982). *A representation for visual information: with application to machine vision*. PhD thesis, Carnegie-Mellon University.
- [Crowley and Parker, 1984] Crowley, J. and Parker, A. (1984). Representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):156–170.
- [Davison et al., 2007] Davison, A., Reid, I., Molton, N., and Stasse, O. (2007). MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, páginas 1052–1067.
- [Davison, 2002] Davison, A. J. (2002). SLAM with a single camera. En *Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots, in conjunction with ICRA*.
- [Davison, 2003] Davison, A. J. (2003). Real-Time Simultaneous Localisation and Mapping with a Single Camera. En *ICCV3*, páginas 1403–1410.

- [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*, 24(7):865–880.
- [DeCarlo and Metaxas, 2000] DeCarlo, D. and Metaxas, D. (2000). Optical Flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38:99–127.
- [DeMenthon and Davis, 1995] DeMenthon, D. F. and Davis, L. S. (1995). Model-based object pose in 25 lines of code. *International Journal in Computer Vision*, 15:123–141.
- [Deriche and Faugeras, 1990] Deriche, R. and Faugeras, O. (1990). Tracking line segments. *Image and Vision Computing*, 8:270–271.
- [Dongarra et al., 1979] Dongarra, J., Moler, C., Bunch, J., and Stewart, G. (1979). *LINPACK Users' Guide SIAM*. Philadelphia, PA.
- [Drummond and Cipolla, 2002] Drummond, T. and Cipolla, R. (2002). Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:932–946.
- [Ess et al., 2007] Ess, A., Neubeck, A., and van Gool, L. (2007). Generalised Linear Pose Estimation. En *British Machine Vision Conference (BMVC '07)*.
- [Fernandez et al., 2007] Fernandez, I., Mazo, M., Lazaro, J., Pizarro, D., Santiso, E., Martin, P., and Losada, C. (2007). Guidance of a mobile robot using an array of static cameras located in the environment. *Autonomous Robots*, 23(4):305–324.
- [Foerstner, 1986] Foerstner, W. (1986). A feature-based correspondence algorithm for image matching. *International Archives of Photogrammetry and Remote Sensing*, 26(3):150–166.
- [Genc et al., 2002] Genc, Y., Riedel, S., Souvannavong, F., and Navab, N. (2002). Marker-less tracking for augmented reality: A learning-based approach. En *International Symposium on Mixed and Augmented Reality*.
- [Gennery, 1992] Gennery, D. (1992). Visual tracking of known three-dimensional objects. *International Journal of Computer Vision*, 7(1):243–270.
- [Haag and Nagel, 1999] Haag, M. and Nagel, H. (1999). Combination of edge element and optical flow estimates for 3-D model-based vehicle tracking intraffic image sequences. *International Journal of Computer Vision*, 35:295–319.
- [Hager and Belhumeur, 1998] Hager, G. and Belhumeur, P. (1998). Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039.
- [Hahnel et al., 2004] Hahnel, D., Burgard, W., Fox, D., Fishkin, K., and Philipose, M. (2004). Mapping and localization with rfid technology. *Proceedings of the IEEE International Conference on Robotics and Automation, 2004.*, 1:1015–1020.

- [Harris, 1992] Harris, C. (1992). *Tracking with Rigid Objects*. MIT Press.
- [Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. En *Fourth Alvey Vision Conference*, Manchester.
- [Hartley and Zisserman, 2003] Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition edition.
- [Hashimoto et al., 2003] Hashimoto, H., Lee, J.-H., and Ando, N. (2003). Self-identification of distributed intelligent networked device in intelligent space. En *ICRA*, páginas 4172–4177.
- [Hazas et al., 2004] Hazas, M., Scott, J., and Krumm, J. (2004). Location-Aware Computing Comes of Age. *Computer*, 37(2):95–97.
- [Heikkila and Silven, 1997] Heikkila, J. and Silven, O. (1997). A Four-Step Camera Calibration Procedure with Implicit Image Correction. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, páginas 1106–1112.
- [Hoff et al., 1996] Hoff, W., Nguyen, K., and Lyon, T. (1996). Computer vision-based registration techniques for augmented reality. *Proceedings of Intelligent Robots and Computer Vision XV, SPIE*, 2904:538–548.
- [Hoover and Olsen, 1999] Hoover, A. and Olsen, B. D. (1999). A Real-Time Occupancy Map from Multiple Video Streams. En *ICRA*, páginas 2261–2266.
- [Hoover and Olsen, 2000] Hoover, A. and Olsen, B. D. (2000). Sensor Network Perception for Mobile Robotics. En *ICRA*, páginas 342–347.
- [Horn et al., 1988] Horn, B., Hilden, H., and Negahdaripour, S. (1988). Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135.
- [Imaging Source, 2008] Imaging Source (2008). Dfk 21bf04 - overview. http://www.theimagingsource.com/en/products/cameras/firewire_color/dfk21bf04/overview/.
- [Jurie and Dhome, 2001] Jurie, F. and Dhome, M. (2001). A simple and efficient template matching algorithm. En *International Conference on Computer Vision*, Vancouver, Canada.
- [Jurie and Dhome, 2002] Jurie, F. and Dhome, M. (2002). Hyperplane approximation for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:996–1000.
- [Kato and Billinghurst, 1999] Kato, H. and Billinghurst, M. (1999). Marker racking and HMD Calibration for a video-based augmented reality conferencing system. En *IEEE and ACM International Workshop on Augmented Reality*.
- [Kato et al., 2000] Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., and Tachibana, K. (2000). Virtual object manipulation on a table-top ar environment. En *International Symposium on Augmented Reality*, páginas 111–119.

- [Ke and Sukthankar, 2004] Ke, Y. and Sukthankar, R. (2004). PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. En *Computer Vision and Pattern Recognition*.
- [Knight et al., 2001] Knight, J., Davison, A., and Reid, I. (2001). Towards constant time slam using postponement. En *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, páginas 406–412.
- [Koller et al., 1993] Koller, D., Daniilidis, K., and Nagel, H. (1993). Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10:257–281.
- [Koller et al., 1997] Koller, D., Klinker, G., Rose, E., Breen, D., Whitaker, R., and Tucceryan, M. (1997). Real-time vision-based camera tracking for augmented reality applications. En *ACM Symposium on Virtual Reality Software and Technology*, páginas 87–94.
- [Kosaka and Nakazawa, 1995] Kosaka, A. and Nakazawa, G. (1995). Vision-based motion tracking of rigid objects using prediction of uncertainties. En *International Conference on Robotics and Automation*, páginas 2637–2644.
- [Kruse and Wahl, 1998] Kruse, E. and Wahl, F. M. (1998). Camera-based Observation of Obstacle Motions to Derive Statistical Data for Mobile Robot Motion Planning. En *ICRA*, páginas 662–667.
- [Lee et al., 2005] Lee, J.-H., Morioka, K., and Hashimoto, H. (2005). Intelligent Space and Mobile Robots. En *The Industrial Information Technology Handbook*, páginas 1–15.
- [Leonard and DurrantWhyte, 1991] Leonard, J. J. and DurrantWhyte, H. F. (1991). Simultaneous map building and localization for an autonomous mobile robot. *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, 3:1442–1447.
- [Li et al., 1993] Li, H., Roivainen, P., and Forchheimer, R. (1993). 3-D motion estimation in model based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:545–555.
- [Li et al., 2004] Li, J., Laganier, R., and Roth, G. (2004). Online Estimation of Trifocal Tensors for Augmenting Live Video. En *ISMAR '04: Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, páginas 182–190, Washington, DC, USA. IEEE Computer Society.
- [Lindeberg, 1994] Lindeberg, T. (1994). Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21:224–270.
- [Liu et al., 1990] Liu, Y., Huang, T. S., and Faugeras, O. D. (1990). Determination of camera location from 2-d to 3-d line and point correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:28–37.
- [Look et al., 2005] Look, G., Kottahachchi, B., Laddaga, R., and Shrobe, H. (2005). A Location Representation for Generating Descriptive Walking Directions. En *IUI '05:*

- Proceedings of the 10th International Conference on Intelligent User Interfaces*, páginas 122–129. ACM Press.
- [Look and Shrobe, 2007] Look, G. and Shrobe, H. (2007). Towards Intelligent Mapping Applications: A Study of Elements Found in Cognitive Maps. En *IUI '07: Proceedings of the 12th International Conference on Intelligent User Interfaces*, páginas 309–312, New York, NY, USA. ACM Press.
- [Lowe, 1992] Lowe, D. G. (1992). Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122.
- [Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. En *International Conference on Computer Vision*, páginas 1150–1157, Greece.
- [Lowe, 2001] Lowe, D. G. (2001). Local feature view clustering for 3d object recognition. En *IEEE Conference on Computer Vision and Pattern Recognition*, páginas 682–688.
- [Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.
- [Lucas and Kanade, 1981] Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. En *International Joint Conference on Artificial Intelligence*, páginas 674–679.
- [Marchand et al., 2001] Marchand, E., Bouthemy, P., and Chaumette, F. (2001). A 2D-3D model-based approach to real-time visual tracking. *Image and Vision Computing*, 19(13):941–955.
- [Matas et al., 2002] Matas, J., Chum, O., Martin, U., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. En *British Machine Vision Conference*, páginas 384–393.
- [Mikolajczyk and Schmid, 2002] Mikolajczyk, K. and Schmid, C. (2002). An affine invariant interest point detector. En *European Conference on Computer Vision*, páginas 128–142. Springer.
- [Montemerlo et al., 2002] Montemerlo, M., Thrun, S., Koller, D., and Wegbreit., B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. En *AAAI National Conference on Artificial Intelligence*, Edmonton, Canada.
- [Moravec, 1977] Moravec, H. (1977). Towards automatic visual obstacle avoidance. En *International Joint Conference on Artificial Intelligence*, página 584.
- [Moravec, 1981] Moravec, H. (1981). *Robot Rover Visual Navigation*. Ann Arbor, Michigan: UMI Research Press.
- [Moreno-Noguer et al., 2007] Moreno-Noguer, F., Lepetit, V., and Fua, P. (2007). Accurate Non-Iterative O (n) Solution to the PnP Problem. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, páginas 1–8.

- [Naimark and Foxlin, 2002] Naimark, L. and Foxlin, E. (2002). Circular data matrix fiducial system and robust image processing for a wearable vision-internal self-tracker. En *International Symposium on Mixed and Augmented Reality*.
- [Nayar et al., 1996] Nayar, S. K., Nene, S. A., and Murase, H. (1996). Real-time 100 object recognition system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1186–1198.
- [Neira and Tardos, 2001] Neira, J. and Tardos, J. (2001). Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robotics and Automation*, 17(6):890–897.
- [Newman and Leonard, 2003] Newman, P. M. and Leonard, J. J. (2003). Consistent Convergent Constant Time SLAM. En *Proceedings of the International Joint Conference on Artificial Intelligence*.
- [Nister and Stewenius, 2007] Nister, D. and Stewenius, H. (2007). A Minimal Solution to the Generalised 3-Point Pose Problem. *Journal of Mathematical Imaging and Vision*, 27(1):67–79.
- [Ocana et al., 2005] Ocana, M., Bergasa, L., Sotelo, M., and Flores, R. (2005). Indoor robot navigation using a POMDP based on WiFi and ultrasound observations. En *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, páginas 2592–2597.
- [Pentland, 1996] Pentland, A. (1996). Smart Rooms. *Scientific American*, 274(4):54–62.
- [Phong et al., 1995] Phong, T., Horaud, R., Yassine, A., and Tao, P. (1995). Object pose from 2-D to 3-D point and line correspondences. *International Journal of Computer Vision*, 15(3):225–243.
- [Pizarro et al., 2005] Pizarro, D., Santiso, E., and Mazo, M. (2005). Simultaneous Localization and Structure Reconstruction of Mobile Robots with External Cameras. En *International Symposium of Industrial Electronics*.
- [Quan and Lan, 1999] Quan, L. and Lan, Z. (1999). Linear N-Point Camera Pose Determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, páginas 774–780.
- [Ravela et al., 1995] Ravela, S., Draper, B., Lim, J., and Weiss, R. (1995). Adaptive tracking and model registration across distinct aspects. En *International Conference on Intelligent Robots and Systems*, páginas 174–180.
- [Rekimoto, 1998] Rekimoto, J. (1998). Matrix: A realtime object identification and registration method for augmented reality. En *Asia Pacific Computer Human Interaction*.
- [Schaffalitzky and Zisserman, 2002] Schaffalitzky, F. and Zisserman, A. (2002). Multi-view matching for unordered image sets, or how do i organize my holiday snaps? *Proceedings of European Conference on Computer Vision*, páginas 414–431.

- [Schmid and Mohr, 1997] Schmid, C. and Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:530–534.
- [Se et al., 2001] Se, S., Lowe, D., and Little, J. (2001). Vision-based Mobile robot localization and mapping using scale-invariant features. En *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, Korea.
- [Shafer et al., 1998] Shafer, S., Krumm, J., Brumitt, B., Meyers, B., Czerwinski, M., and Robbins, D. (1998). The New EasyLiving Project at Microsoft Research. En *Proceedings, Joint DARPA/NIST Smart Spaces Workshop*.
- [Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good Features to Track. En *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, páginas 593 – 600.
- [Simon et al., 2000] Simon, G., Fitzgibbon, A., and Zisserman, A. (2000). Markerless tracking using planar structures in the scene. En *International Symposium on Mixed and Augmented Reality*, páginas 120–128.
- [Skrypnik and Lowe, 2004] Skrypnik, I. and Lowe, D. G. (2004). Scene modelling, recognition and tracking with invariant image features. En *International Symposium on Mixed and Augmented Reality*, páginas 110–119.
- [Smith and Brady, 1995] Smith, S. M. and Brady, J. M. (1995). Susan - a new approach to low level image processing. Technical report, Oxford University.
- [State et al., 1996] State, A., Hirota, G., Chen, D., Garrett, W., and Livingston, M. (1996). Superior augmented reality registration by integrating landmark tracking and magnetic tracking. *Computer Graphics, SIGGRAPH Proceedings*, páginas 429–438.
- [Stein and Hebert, 2005] Stein, A. and Hebert, M. (2005). Incorporating Background Invariance into Feature-Based Object Recognition. *Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05)*, 1:37–44.
- [Steinhaus et al., 1999] Steinhaus, P., Ehrenmann, M., and Dillmann, R. (1999). MEP-HISTO. A Modular and Extensible Path Planning System Using Observation. *Lecture Notes in Computer Science*, 1542:361–375.
- [Steinhaus et al., 2007] Steinhaus, P., Strand, M., and Dillmann, R. (2007). Autonomous robot navigation in human-centered environments based on 3d data fusion. *EURASIP Journal on Applied Signal Processing*, 2007(1):224–224.
- [Stewenius et al., 2005] Stewenius, H., Oskarsson, M., and Aström, K. (2005). Reconstruction from Planar Motion Image Sequences with Applications for Autonomous Vehicles. *Lecture notes in computer science*, 3540:609–618.
- [Tomasi and Kanade, 1991] Tomasi, C. and Kanade, T. (1991). Detection and Tracking of Point Features. Technical Report CMU-CS-91-132, Carnegie Mellon University.

- [Tordoff et al., 2002] Tordoff, B., Mayol, W., de Campos, T., and Murray, D. (2002). Head pose estimation for wearable robot control. En *British Machine Vision Conference*, páginas 807–816.
- [Triggs et al., 2000] Triggs, B., McLauchlan, P., Hartley, R., and Fitzgibbon, A. (2000). Bundle Adjustment – A Modern Synthesis. En *Vision Algorithms: Theory and Practice*, volumen 1883, páginas 298–372. Springer-Verlag.
- [Tsai, 1986] Tsai, R. (1986). An efficient and accurate camera calibration technique for 3d machine vision. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 374.
- [T.Sogo et al., 1999] T.Sogo, H.Ishiguro, and T.Ishida (1999). Acquisition of Qualitative Spatial Representation by Visual Observation. En *IJCAI99*.
- [Tuytelaars and VanGool, 2000] Tuytelaars, T. and VanGool, L. (2000). Wide baseline stereo matching based on local affinity invariant regions. En *British Machine Vision Conference*, páginas 412–422.
- [Uenohara and Kanade, 1991] Uenohara, M. and Kanade, T. (1991). Vision-based object registration for real-time image overlay. *Journal of Cognitive Neuroscience*, 3:71–86.
- [Vacchetti et al., 2004] Vacchetti, L., Lepetit, V., and Fua, P. (2004). Stable real-time 3d tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1385–1391.
- [Van der Merwe et al., 2000] Van der Merwe, R., de Freitas, N., Doucet, A., and Wan, E. (2000). The Unscented Particle Filter. Technical Report CUED/F-INFENG/TR380, Cambridge University Engineering Department.
- [Villadangos et al., 2005] Villadangos, J. M., Urena, J., Mazo, M., Hernandez, A., Alvarez, F. J., Garcia, J. J., Marziani, C. D., and Alonso, D. (2005). Improvement of Ultrasonic beacon-based Local Position System Using Multi-Access Techniques. En *WISP05*, página CDROM Edition.
- [Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. En *Conference on Computer Vision and Pattern Recognition*, páginas 511–518.
- [Weiser, 1993a] Weiser, M. (1993a). PARC builds a world saturated with computation. *Science (AAAS)*, páginas 3–11.
- [Weiser, 1993b] Weiser, M. (1993b). Some Computer Science Problems in Ubiquitous Computing. *Communications of the ACM*.
- [Weiser, 1999] Weiser, M. (1999). The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11.
- [Xu et al., 2007] Xu, J., Ye, G., and Zhang, J. (2007). Long-term Trajectory Extraction for Moving Vehicles. *Multimedia Signal Processing, 2007. MMSP 2007. IEEE 9th Workshop on*, páginas 223–226.

- [Zhang, 1999] Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. *International Conference on Computer Vision*, 1:666–673.
- [Zhang et al., 1995] Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q. (1995). A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intel ligence*, 78:87–119.