



**DEPARTAMENTO  
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

# Localización externa para robots móviles utilizando múltiples cámaras

---

22 de diciembre de 2016

Bruno Alejandro Juffar  
brunojuffar@gmail.com

Directores

Nitsche, Matias A.  
mnitsche@dc.uba.ar

Co-Directores

Pessacg, Facundo  
fpessacg@dc.uba.ar



**Facultad de Ciencias Exactas y Naturales**

Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (+54 +11) 4576-3300

<http://www.exactas.uba.ar>

# Índice general

<b>1. Introducción</b>	<b>3</b>
1.1. Estado del arte . . . . .	4
1.2. Objetivos . . . . .	7
1.3. Organización de la tesis . . . . .	8
<b>2. Conceptos preliminares</b>	<b>9</b>
2.1. Modelo de cámara . . . . .	9
2.2. WhyCon . . . . .	11
2.2.1. Descripción del método . . . . .	12
2.3. Localización probabilística . . . . .	13
2.3.1. Filtro Bayesiano . . . . .	14
2.3.2. Filtro de Kalman . . . . .	15
2.3.2.1. Fase de predicción . . . . .	16
2.3.2.2. Fase de actualización . . . . .	16
2.3.3. Filtro de Kalman Extendido . . . . .	17
<b>3. Método</b>	<b>19</b>
3.1. Calibración multi-cámara . . . . .	19
3.1.1. WhyCon . . . . .	20
3.1.1.1. Funcionamiento 3D . . . . .	21
3.1.1.2. Funcionamiento 2D . . . . .	22
3.1.2. Kalibr . . . . .	22
3.2. Detalle del método . . . . .	24
3.2.1. Estado . . . . .	25
3.2.2. Modelo de movimiento . . . . .	26
3.2.2.1. Velocidad obtenida por odometría . . . . .	27
3.2.2.2. Velocidad estimada a partir de observaciones . . . . .	28
3.2.3. Modelo de observaciones . . . . .	28

<b>4. Experimentos</b>	<b>31</b>
4.1. Calibración . . . . .	32
4.1.1. Kalibr . . . . .	33
4.1.2. WhyCon . . . . .	37
4.2. Modelo de error en función de la distancia . . . . .	38
4.3. Ajuste del parámetro de filtrado de la velocidad . . . . .	40
4.4. Análisis de precisión: máximo solapamiento con dos cámaras . . . . .	42
4.5. Análisis de precisión: máximo solapamiento con cuatro cámaras . . . . .	46
4.6. Análisis de precisión: máxima cobertura con cuatro cámaras . . . . .	52
4.7. Análisis de precisión: escena general . . . . .	56
4.8. Análisis de casos particulares . . . . .	59
<b>5. Conclusiones</b>	<b>63</b>
5.1. Trabajo futuro . . . . .	65

# Capítulo 1

## Introducción

Uno de los problemas fundamentales en el campo de la robótica móvil es el de la localización, que consiste en determinar la pose del robot, es decir posición y orientación, respecto de un sistema de coordenadas determinado. Un robot autónomo, para poder navegar dentro de un entorno dado, debe contar con dicha información. En estos casos, el robot debe localizarse en tiempo real. Esto significa que, además de obtener una localización precisa, el sistema debe obtener esta información en un intervalo de tiempo suficientemente bajo como para asegurar su correcto funcionamiento.

Otro contexto en el cual debe resolverse la localización de un robot es durante la realización de experimentos en los cuales se busca contar con la información de la pose real (idealmente) del robot. El objetivo en este caso es el de contrastar dicha información con la obtenida mediante algún otro método de localización, generalmente interno, que se está evaluando. En este sentido, la información de la pose del robot que se busca obtener se utiliza como información de verdad absoluta (*ground truth*). Por esta razón y a diferencia del caso anterior, no necesariamente se busca resolver la localización en tiempo real ya que se pueden procesar los datos sin restricciones temporales en una etapa posterior.

En general, los métodos de localización pueden ser categorizados según su enfoque en relativos o absolutos, o bien en internos o externos.

Un método de localización relativa estima la pose del robot en forma acumulativa a partir de integrar la información del movimiento realizado en el último intervalo de tiempo respecto de la pose anterior, utilizando diversos sensores. Dado que la estimación de movimiento relativo siempre introduce, en mayor o menor medida un error de estimación, el error de localización crece en forma no acotada bajo este enfoque. Por ejemplo, la odometría es una forma de localización relativa y consiste en el uso de *encoders* que permiten sensar los giros de las ruedas de un robot y traducir los mismos a un movimiento de este último.

Por otro lado, los métodos de localización absoluta utilizan generalmente sensores con los cuales ante cada medición es posible obtener la pose absoluta del robot, independientemente de sensados previos. Por lo tanto, en estos casos el error no es acumulativo. En general con este enfoque se requiere de alguna infraestructura preexistente que resuelva la localización. Un ejemplo ampliamente conocido



de un sistema de localización absoluta es el del GPS (*Global Positioning System*), el cual utiliza un conjunto de satélites en órbita que permiten estimar la posición del dispositivo receptor.

Otra clasificación posible de los métodos de localización distingue enfoques internos de externos, según si los sensores se encuentran a bordo del robot o si observan a este desde una ubicación externa. En los casos en donde un robot debe operar en un entorno desconocido de antemano, por ejemplo un robot que debe explorar un área, resulta necesario basarse en los propios sensores del robot (como los encoders) dado que no puede asumirse una infraestructura preexistente. En este caso se estaría hablando de un sistema de localización interna. Por el contrario, si para localizar al robot se cuenta con algún sistema como el del GPS, podemos decir que el método utilizado es de localización externa.

De esta forma, es común asociar a los métodos de localización relativa con los métodos de localización interna, por la utilización de sensores a bordo del robot para estimar el movimiento relativo. Asimismo, los métodos de localización absoluta se suelen asociar con métodos de localización externa por la utilización de sensores externos al robot. En esta tesis el foco estará puesto en los sistemas de localización externa.

Cabe aclarar que, si bien el sistema GPS parecería ser una buena solución al problema de la localización externa, este tiene algunas limitaciones importantes. Por un lado, solo puede ser utilizado en un entorno exterior donde pueda recibirse la señal correspondiente. Por otro, el sistema GPS no suele ser lo suficientemente preciso para la navegación a corto plazo.

Como alternativa, y particularmente en los casos en donde el entorno de operación del robot es conocido de antemano y este puede ser intervenido, por ejemplo en una fábrica o en un ambiente de experimentación controlado, resulta atractivo el desarrollo de un sistema de localización externa que no se vea afectado por esas limitaciones, como el que se presenta en esta tesis.

## 1.1. Estado del arte

Los diversos métodos de localización externa propuestos en la literatura y existentes en la actualidad varían principalmente según el tipo de sensado utilizado. En este sentido pueden distinguirse sistemas basados en sensado *activo*, donde se emite una señal específica que se espera detectar, o en sensado *pasivo*, donde se captura información existente del ambiente. En el caso del sensado activo se usan señales de radio, luz o sonido. La dificultad de este tipo de sensado es que suelen aparecer problemas de interferencia entre las distintas señales utilizadas u otras existentes, por ejemplo la luz solar afectando señales lumínicas. Un ejemplo de un sistema basado en sensado activo es [1], en donde se utilizan emisores LED como marcadores del objeto a localizar y una cámara monocular para identificar a dichos marcadores. Mediante un proceso de correspondencia 2D-3D, se obtienen las posiciones de los marcadores y, por ende, la pose del objeto en cuestión. Aunque los experimentos de este trabajo arrojaron buenos resultados, los autores señalan que para los escenarios de iluminación no controladas, los sistemas de localización pasiva parecen ser más adecuados. Otro ejemplo de este

tipo de sistemas es NorthStar [2], el cual utiliza proyecciones en el techo para marcar el ambiente de forma no permanente. Al proyectar un patrón conocido, la posición de una cámara observando el techo se puede obtener por reproyección.

Por las limitaciones previamente mencionadas y debido a que el costo de los sistemas basados en sensado activo suele ser mayor, se han propuesto diversos métodos de localización externa utilizando sensado pasivo mediante el uso de cámaras de video convencionales. Estas observan una escena conteniendo algún tipo de patrón a seguir que pueda ser identificado visualmente. Estos sistemas se basan fuertemente en métodos de reconocimiento de patrones y visión computacional, originalmente utilizados para aplicaciones de realidad aumentada, que permiten recuperar la posición y orientación del patrón. Ejemplos de algunos sistemas que utilizan técnicas de sensado pasivo son ARTag [3], ARTagToolKit+ [4], ambos basados en ARTagToolKit [5]. Dichos sistemas son ampliamente utilizados como sistemas de localización de robots móviles, ya sean para obtener la estimación de la pose en tiempo real [3, 6], o para obtener información de *ground truth* [7].

Otro sistema de localización externa basado en técnicas de sensado pasivo es Apriltag [8], de conceptos similares a ARTag, el cual identifica un tipo de patrón en la imagen como el que se observa en la figura 1.1a. Estos patrones, permiten codificar no solo la orientación sino también un identificador único. Apriltag es flexible y robusto, por este motivo muchos sistemas lo utilizan como base de su desarrollo con el fin de obtener la estimación de la pose [9, 10, 11], y muchos otros para obtener información de *ground truth* [12, 13, 14].

Siguiendo en esta línea, otro sistema de localización externa basado en sensado pasivo es TRIP [15], en donde el patrón consta de un conjunto de anillos concéntricos, divididos en varias regiones angulares, cada una de las cuales puede ser negra o blanca, como se observa en en la figura 1.1b. El esquema de codificación, que incluye la comprobación de paridad, permite que el método TRIP distinga entre  $3^9$  patrones distintos.

Entre los diversos métodos de localización externa basados en la detección de patrones visuales, podemos distinguir el método conocido como WhyCon [16], el cual se presenta como una alternativa de bajo costo a sistemas comerciales tales como ViCon [17]. El sistema WhyCon es capaz de detectar patrones circulares monocromáticos, que pueden ser impresos fácilmente en una impresora convencional (ver figura 1.1c). A partir del conocimiento de las dimensiones reales de dicho patrón, es posible recuperar la posición del mismo en tres dimensiones utilizando una sola cámara. Este tipo de patrón en particular permite emplear algoritmos de visión por computadora muy eficientes que resultan en tasas de detección de cientos a miles de cuadros por segundo, utilizando hardware convencional, valores incluso muy superiores a las capacidades de las cámaras de video convencionales actuales. Además, la precisión de dicho sistema se encuentra en el orden del 1 al 3% de error relativo.

Los sistemas anteriormente mencionados, al contar con una única vista de la escena, poseen limitaciones tales como un área de cobertura definida directamente por el alcance de la cámara y una precisión mayormente limitada por la ubicación y características de la misma. Por esta razón resulta interesante analizar sistemas de localización basados en el uso de múltiples cámaras. Por un lado,

esto permite aumentar el área de trabajo posicionando las cámaras minimizando el solapamiento entre lo que ve cada una, y por otro, es posible mejorar la precisión de la localización maximizando el solapamiento utilizando distintas cámaras convergentes, logrando así que varias cámaras vean el mismo patrón a localizar desde diversos puntos de vista. De hecho, es posible combinar ambas estrategias según se requiera mayor área de cobertura o mayor precisión en cada sector del ambiente.

En la actualidad, existen algunos métodos de localización externa multi-cámara que pueden mencionarse. Por ejemplo, en [18] se presenta un conjunto de datos (*dataset*) adquiridos mediante un robot móvil desplazándose en diversos entornos y con múltiples sensores. Como parte del método utilizado para obtener la información de pose de *ground truth* incluida en el *dataset*, se utiliza un conjunto de cámaras fijas que observan la escena, con el objetivo de detectar una serie de patrones visuales adosados al robot. En particular, las cámaras están colocadas con un solapamiento mínimo o incluso sin solaparse dado que se buscó maximizar el área de cobertura, sin centrarse en mejorar la estimación de la pose con respecto a sistemas con una única cámara. Dado que cada cámara permite obtener un estimado de la pose del patrón desde su punto de vista, es necesario contar con algún método de fusión de datos para poder unificar dicha información y obtener un único estimado de la pose del patrón. En particular, en este trabajo se utiliza el filtro conocido como *Extended Kalman Smoother* [19], similar al Filtro de Kalman Extendido [20], pero que realiza correcciones no solamente en la estimación actual sino también sobre las estimaciones de pasos anteriores. Esto tiene sentido en este contexto ya que el trabajo se basa en obtener información de *ground truth*.

En otro trabajo [21], en cambio, la pose del robot no es obtenida mediante la detección de un patrón específico, sino que se detecta al robot directamente en las imágenes a partir de diferenciar las regiones en movimiento de un escenario que se asume estático. Esto es posible debido a que el entorno en el cual se trabaja es controlado, o sea que es preparado para utilizar este método en particular. El método propuesto consiste en minimizar una función objetivo de correspondencias 2D-3D. El aporte interesante de este trabajo es la generalización de una función objetivo, planteada originalmente para una sola cámara a varias. Cabe aclarar que en este trabajo no se aborda el problema de hallar la pose relativa entre las distintas cámaras.

En general, si bien se pueden identificar trabajos en los cuales se proponen métodos de localización externa basados en el uso de múltiples cámaras, no existe en la actualidad un sistema de propósito general que resuelva este problema dado que generalmente se proponen sistemas *ad-hoc*. Por otro lado, entre los sistemas de localización monoculares, se puede destacar WhyCon [16] por su precisión, eficiencia y robustez. Por estas razones, resulta altamente atractiva la generalización del sistema WhyCon al caso con múltiples cámaras, con el objetivo de satisfacer las limitaciones mencionadas y así obtener un sistema de localización multi-cámara de propósito general.

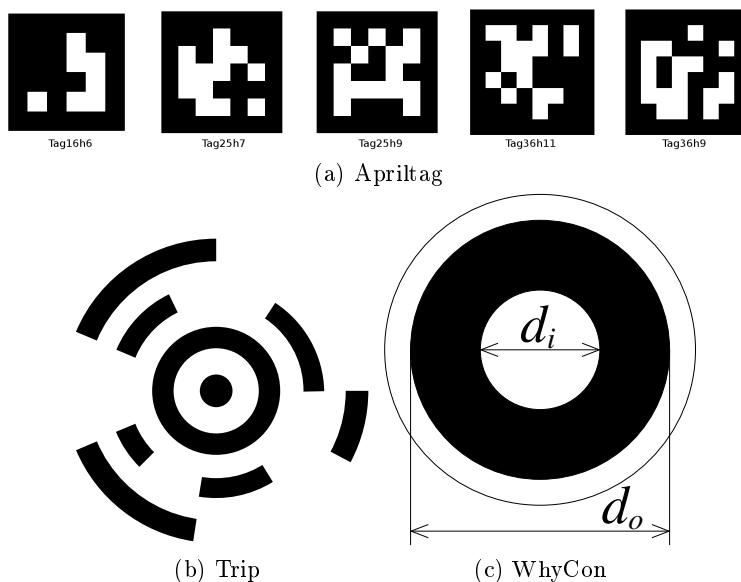


Figura 1.1: Patrones.

## 1.2. Objetivos

El objetivo central de este trabajo es el desarrollo de un sistema de localización multi-cámara, fusionando la información de múltiples sistemas de localización monoculares y opcionalmente de sensores a bordo del robot, obteniendo así una única estimación de la pose del mismo. Con esto lo que se busca lograr es ampliar el área de cobertura como así también incrementar la precisión de la estimación, según la disposición de las cámaras utilizadas. En esta tesis se propone el uso del sistema WhyCon como sistema de localización monocular.

Se espera que el método de fusión que se proponga sea capaz de resolver las diversas configuraciones de cámaras posibles a utilizar con dicho sistema, con y sin solapamiento entre vistas, arrojando una única estimación de mayor precisión que las estimaciones individuales obtenibles por cada cámara en particular o permitiendo incrementar el área de cobertura del sistema de localización.

Por último, en los casos en donde existan sensores odométricos o inerciales a bordo del robot, se busca que el sistema sea capaz de integrar también esta información para mejorar las estimaciones a corto plazo.

Como objetivo final, se espera obtener una implementación del método de localización propuesto que sea utilizable en los contextos de uso planteados anteriormente de forma simple y que sea fácilmente integrable a otros sistemas existentes. Por esta razón, se plantea el desarrollo en base al framework ROS (Robot Operating System) [22], ampliamente utilizado en el campo de la robótica.

### 1.3. Organización de la tesis

En este capítulo se introdujo la problemática de la localización y se planteó la motivación para la solución propuesta en esta tesis. En el siguiente capítulo se describe el marco teórico necesario para comprender el método presentado y se amplían conceptos introducidos al momento tales como técnicas de fusión y localización probabilística y el funcionamiento del sistema WhyCon, en el cual se basa el método propuesto. En el tercer capítulo se presenta detalladamente el método específico de localización multi-cámara propuesto en esta tesis, buscando cumplir con los distintos objetivos mencionados. El cuarto capítulo se enfoca en analizar el desempeño del método propuesto, a partir de los resultados obtenidos mediante los diversos experimentos realizados. Por último, en el quinto capítulo se presentan las conclusiones que se obtuvieron y se mencionan las posibles mejoras y aspectos que podrían introducirse al método propuesto en una futura instancia.

## Capítulo 2

# Conceptos preliminares

### 2.1. Modelo de cámara

Una cámara puede ser modelada como un mapeo, o una proyección, entre el mundo 3D y una imagen en 2D [23]. En la figura 2.1, se observa como un punto en el espacio 3D es proyectado al plano de la imagen, o plano focal.

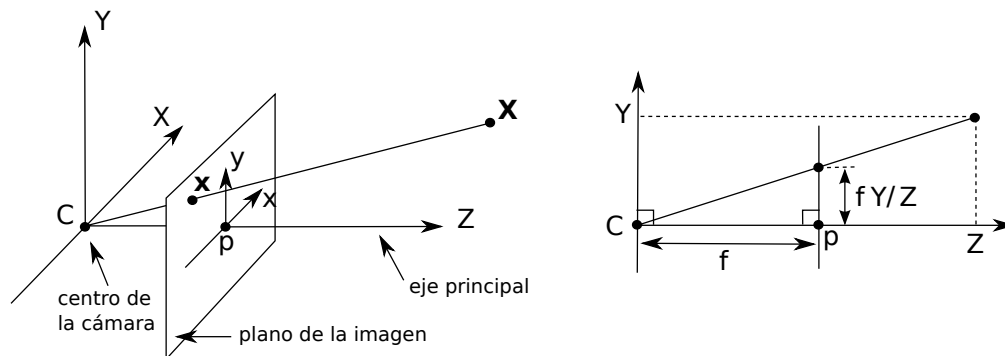


Figura 2.1: Modelo geométrico de cámara *pinhole*.  $C$  representa al centro de la cámara y  $p$  al punto principal en la imagen. El centro de la cámara se ubica en el origen de coordenadas.

Utilizando coordenadas homogéneas, si  $\mathbf{x} \in \mathbb{R}^3$  la representación del punto en la imagen y el vector  $\mathbf{X} \in \mathbb{R}^4$  el punto en las coordenadas del mundo, se define  $\mathbf{P} \in \mathbb{R}^{4 \times 3}$  como la matriz de proyección que permite proyectar cualquier punto del espacio en coordenadas del mundo al plano de la imagen, tal que:

$$\mathbf{x} = \mathbf{P} \mathbf{X}$$

donde  $\mathbf{P}$  se define como:

$$\mathbf{P} = \mathbf{K} [\mathbf{R} | \mathbf{t}]$$

y donde  $\mathbf{K} \in \mathbb{R}^{3 \times 3}$  es la matriz de parámetros intrínsecos definida como:

$$\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

donde  $f$  corresponde a la longitud focal y  $p_x, p_y$  al punto principal (ver figura 2.1). Por otro lado, debido a que la cámara está asociada a un sistema de coordenadas  $C$ , la posición y orientación de dicho sistema respecto del sistema  $W$  del mundo se describe mediante la rotación  $\mathbf{R}$  y traslación  $\mathbf{t}$  que los relaciona, que se conocen como parámetros extrínsecos (figura 2.2).

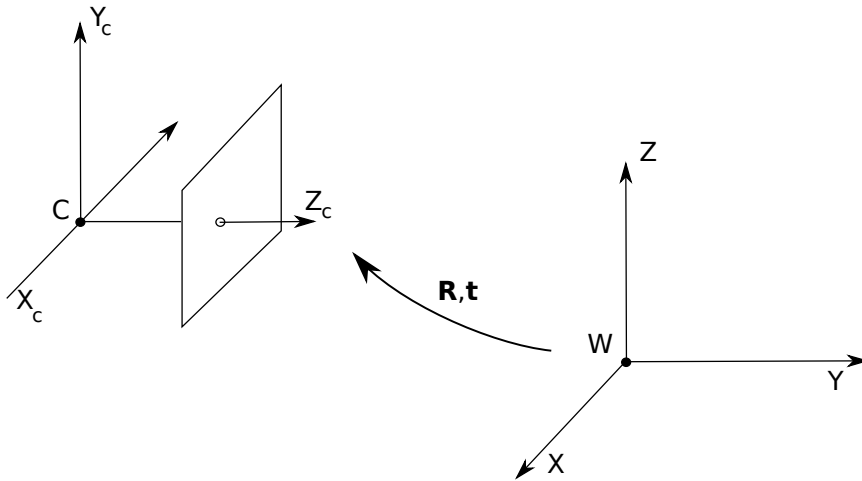


Figura 2.2: Transformación euclidiana entre el mundo y la cámara.

A continuación se detallan los pasos a seguir para obtener dicha proyección.

Partiendo desde la figura 2.1, siendo  $f$  la representación del plano focal, se considera  $z = f$  al plano de la imagen, y el punto en el espacio  $\mathbf{X} = (x, y, z)^T$  es mapeado al punto  $\mathbf{x} = (f x/z, f y/z, f)^T$  en dicho plano. Entonces:

$$(x, y, z)^T \mapsto (f x/z, f y/z)^T$$

describe el mapeo entre el mundo y las coordenadas de la imagen. Este es un mapeo entre el espacio euclidiano en  $\mathbb{R}^3$  y el espacio euclidiano en  $\mathbb{R}^2$ .

Esta expresión asume que el punto principal está en el origen de coordenadas, lo cual en la práctica suele no suceder. Por lo tanto, en general, el mapeo queda de la siguiente forma:

$$(x, y, z)^T \mapsto (f x/z + p_x, f y/z + p_y)^T$$

convenientemente, esta expresión se puede ver en coordenadas homogéneas como:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f x + z p_x \\ f y + z p_y \\ z \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

en donde se puede observar la matriz  $\mathbf{K}$  (ver ecuación 2.1) utilizada para dicho mapeo.

Por otro lado, si  $X$  representa el punto en el mundo ( $W$ ) y  $X_{cam}$  el mismo punto en las coordenadas de la cámara, entonces:

$$X_{cam} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{C} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{C} \\ 0 & 1 \end{bmatrix} X$$

donde  $\tilde{C}$  representa el centro de la cámara en las coordenadas del mundo, y  $\mathbf{R} \in \mathbb{R}^{3 \times 3}$  la matriz de rotación en el marco de coordenadas de la cámara.

Uniéndolo todo se obtiene que:

$$\mathbf{x} = \mathbf{K} \mathbf{R} \begin{bmatrix} \mathbf{I} & -\tilde{C} \end{bmatrix} X$$

Es conveniente no instanciar explícitamente el centro de la cámara y tomar como la transformación del mundo a la cámara a  $X_{cam} = \mathbf{R}X + t$ , con  $t = -\mathbf{R}\tilde{C}$ . De esta forma queda planteada la matriz de proyección del siguiente modo:

$$\mathbf{P} = \mathbf{K} [\mathbf{R} | t]$$

## 2.2. WhyCon

WhyCon es un sistema basado en visión de localización de robots móviles. El componente principal del sistema es un eficiente algoritmo de detección de patrones en blanco y negro. Utilizando cámaras de bajo costo, este algoritmo es capaz de procesar cientos de imágenes por segundo, realizando el seguimiento de cientos de objetos, con una precisión elevada y una gran robustez en cuanto a las variables de iluminación y la perspectiva del patrón. Por estos motivos en muchos trabajos se ha utilizado este sistema como un módulo de localización en tiempo real [24, 25, 26, 27], y muchos otros lo utilizan como fuente de información de ground truth [28, 29, 30]



### 2.2.1. Descripción del método

El método de WhyCon para resolver la localización se basa en la detección rápida y precisa de un patrón circular, blanco y negro, que consiste en un anillo negro sobre un fondo blanco (ver figura 2.3).

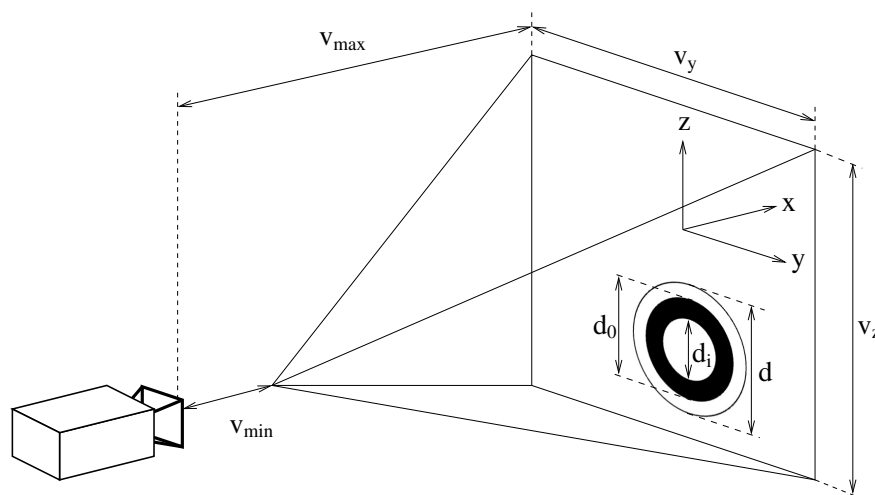


Figura 2.3: Geometría del espacio operacional.

En una primera etapa, se resuelve la detección del patrón para identificar su posición en coordenadas de píxeles. El algoritmo recorre la imagen de izquierda a derecha, de arriba hacia abajo, clasificando los píxeles en blancos y en negros, según un cierto umbral que se determina en forma adaptativa. Ubicando un primer píxel negro, el algoritmo comienza a recorrer los alrededores del píxel identificando el posible patrón. Una vez que se detecta el patrón en la imagen y este pasa todas las pruebas que confirman que realmente es el patrón en cuestión (por ejemplo circularidad), se obtiene el centro del mismo a partir de calcular el centroide de las coordenadas de los píxeles que conforman la región detectada.

Como segundo paso, se buscan recuperar las coordenadas 3D del mismo en el mundo. A partir del conocimiento de que todo círculo se proyecta como una elipse en las imágenes de una cámara y de los modelos proyectivos correspondientes, se utilizan los parámetros de la elipse detectada (dirección y tamaño de los semiejes) para recuperar la posición del patrón en coordenadas 3D respecto del sistema de coordenadas.

Para ello, se calcula primero la dirección del rayo que une el centro de la elipse en el plano de la imagen con el centro de la cámara. Luego, la distancia del patrón sobre este rayo se obtiene a partir del conocimiento de las dimensiones reales del patrón en el mundo (diámetro externo, diámetro interno) y de la relación entre éstas y las de la elipse (medidas en píxeles). Intuitivamente, a mayor distancia al patrón, menor cantidad de píxeles tendrá la elipse en la imagen.

Debido a esta forma de estimación de la dirección del rayo y la distancia sobre el mismo a la que se encuentra el patrón, la principal fuente de error de WhyCon proviene de esta última variable.

Esto se debe a que, por un lado, a medida que el tamaño en píxeles del patrón es cada vez menor (por ejemplo, a medida que se lo aleja de la cámara o se utiliza una menor resolución) la distancia que representa una diferencia de tan solo un píxel es cada vez mayor. Por esta razón, teniendo en cuenta el ruido que poseen las imágenes capturadas por cualquier cámara, puede verse como esto afectaría considerablemente la estimación de la distancia cuando la cantidad de píxeles del patrón sea baja. Por otra parte, la precisión con la cual se puede obtener la dirección del rayo de proyección al patrón depende únicamente de la resolución de la cámara (debido al cálculo del mismo a partir del centroide de los píxeles correspondientes) y no de la distancia.

Debe recalarse que al utilizar un patrón circular simétrico, WhyCon no es capaz de estimar la orientación del patrón de forma inequívoca sin información adicional. En consecuencia, al utilizar un único patrón como el de la figura 2.3, solo será posible obtener la posición del mismo.

Hasta aquí se describe la detección de un único patrón. Sin embargo, este sistema de localización es capaz de detectar varios patrones simultáneamente, simplemente ejecutando el algoritmo de detección varias veces en forma consecutiva. El método recuerda los patrones ya reconocidos para no volver sobre estos en el transcurso de la ejecución. De todas formas, el método así propuesto no es capaz de distinguir dos patrones iguales en forma unívoca. Por esta razón, en este trabajo se trabajará con un solo patrón, dejando para un trabajo posterior resolver la localización con varios patrones.

### 2.3. Localización probabilística

Al estimar la ubicación de un robot existirá siempre un grado de incertidumbre sobre el valor real de dicha información. Por esta razón, la localización nunca puede ser considerada exacta. Esto sucede por varios motivos. Por un lado, esto se debe a que la percepción del mundo es inherentemente ambigua, y por el otro, a que el robot percibe el entorno mediante sensores que no son perfectos. De esta forma, la información recibida puede tener un grado alto de exactitud, pero nunca será ciento por ciento confiable. Por lo tanto se suele utilizar un modelo probabilístico de la ubicación del robot, es decir, se estima la probabilidad  $p(X)$  de que el robot este en la posición  $X$ .

En el contexto de la localización, en vez de hablar únicamente de la pose del robot, se introduce la noción del estado, al que se lo puede describir como el conjunto de todos los aspectos del robot que son relevantes al problema de la localización. El estado puede incluir, además de la información de su posición y orientación en un momento determinado, la información de velocidad, aceleración, etc. El estado se suele denotar como  $\mathbf{x}_t$  debido a que tendrá un valor determinado en un tiempo  $t$  e irá cambiando de acuerdo a cierto modelo.

Existen dos tipos fundamentales de interacciones entre el robot y su entorno. El robot puede influir, modificar su estado a través de sus actuadores y, además, puede recolectar información de sensores que observan el entorno en el que se encuentra.

Como se mencionó anteriormente, la evolución del estado se rige por un modelo que también

puede ser descrito en forma probabilística. En principio, el estado actual esta condicionado por las anteriores estimaciones así como por la historia de acciones y observaciones realizadas. Por lo tanto, se puede representar la pose desconocida del robot como una función de probabilidad de la siguiente forma:

$$p(\mathbf{x}_t | \mathbf{x}_0, \dots, \mathbf{x}_{t-1}, \mathbf{z}_1, \dots, \mathbf{z}_{t-1}, \mathbf{u}_1, \dots, \mathbf{u}_{t-1})$$

donde  $\mathbf{z}_t$  y  $\mathbf{u}_t$  corresponden a las observaciones de los sensores y acciones realizadas por los actuadores del robot en tiempo  $t$ , respectivamente.

Otro concepto clave es el de creencia (*belief*). Debido a que en la realidad el estado  $\mathbf{x}_t$  no es directamente observable, sino que el robot únicamente puede obtener información del entorno mediante sensores, se introduce el concepto de *belief*, definido como:

$$bel(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{z}_1, \dots, \mathbf{z}_t, \mathbf{u}_1, \dots, \mathbf{u}_t)$$

en donde se puede notar que este cálculo se realiza una vez que se obtiene la observación  $\mathbf{z}_t$ , por lo que  $bel(\mathbf{x}_t)$  representa el *posterior belief*. En ese sentido, se puede distinguir  $bel(\mathbf{x}_t)$  de  $\overline{bel}(\mathbf{x}_t)$  conocido como *prior belief*, que representa el conocimiento de  $\mathbf{x}_t$  previo a incorporar el sensado actual  $\mathbf{z}_t$ :

$$\overline{bel}(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{z}_1, \dots, \mathbf{z}_{t-1}, \mathbf{u}_1, \dots, \mathbf{u}_t)$$

Por esta razón,  $\overline{bel}(\mathbf{x}_t)$  se suele conocer en el contexto de los filtros probabilísticos como *predicción*. Por otra parte,  $bel(\mathbf{x}_t)$  es conocida como la *corrección* o *actualización* ya que incluye la última observación  $\mathbf{z}_t$  para mejorar la estimación predicha.

### 2.3.1. Filtro Bayesiano

El algoritmo más general para calcular *beliefs* es el del Filtro de Bayes [20]. En el contexto del problema de la localización, permite estimar la pose de un robot, que cuenta con sensores y actuadores, con respecto a un sistema de coordenadas dado.

---

**Algoritmo 2.1** Filtro Bayesiano.

---

**Algorithm** `Bayes_Filter`( $bel(\mathbf{x}_{t-1}), \mathbf{u}_t, \mathbf{z}_t$ ) :

for all  $\mathbf{x}_t$  do

$$\overline{bel}(\mathbf{x}_t) = \int p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1}) bel(\mathbf{x}_{t-1}) dx_{t-1}$$

$$bel(\mathbf{x}_t) = \eta p(\mathbf{z}_t | \mathbf{x}_t) \overline{bel}(\mathbf{x}_t)$$

end for

return  $bel(\mathbf{x}_t)$

---

Como se puede observar en el algoritmo 2.1, el Filtro Bayesiano es un algoritmo recursivo que

utiliza la creencia anterior  $bel(\mathbf{x}_{t-1})$ , la observación actual  $\mathbf{z}_t$  y la información del movimiento actual  $\mathbf{u}_t$  para calcular la creencia actual  $bel(\mathbf{x}_t)$ .

La derivación del filtro se basa en una premisa fundamental sin la cual no sería posible obtener la estimación final. Si se observa el algoritmo, se puede ver que la creencia en tiempo  $t$  es calculada en base a únicamente información del tiempo anterior  $t - 1$ . Esto es posible si se trabaja bajo la hipótesis de Markov, que postula que los datos pasados y futuros son independientes si se conoce el estado actual  $\mathbf{x}_t$ . En este caso, se dice que el estado es *completo*.

Si bien esta hipótesis resulta razonable en la mayoría de los contextos de aplicación, podrían existir violaciones a dicha asunción como por ejemplo movimientos no modelados en el estado (una persona empujando el robot) o por ejemplo personas que interfieran en la información captada por los sensores. Sin embargo, el Filtro de Bayes es lo suficientemente robusto en la práctica si el estado está bien definido, o sea que se lo puede asumir completo.

Existen varias técnicas de estimación basadas en este filtro, donde una de las más utilizadas y conocidas es el Filtro de Kalman (*KF*) y su versión extendida (*EKF*). Este último es el propuesto para la fusión de la información que se debe realizar como parte del sistema de localización multi-cámara que se presenta en esta tesis.

### 2.3.2. Filtro de Kalman

El filtro de Kalman [31] fue propuesto en la década del 50 por Rudolph Emil Kalman y probablemente sea la técnica más estudiada para implementar Filtros Bayesianos. Es un método recursivo de filtrado y predicción en sistemas lineales de estados continuos. El filtro utiliza las observaciones de los sensores en el tiempo para realizar estimaciones, sabiendo que estas variables tienen un cierto ruido y son inexactas.

El Filtro de Kalman es un estimador que utiliza únicamente el estado anterior, a diferencia de otras técnicas más sofisticadas (y generalmente computacionalmente costosas) que utilizan todos los estados anteriores para poder estimar el actual, tales como *Bundle Adjustment* [32]. Por esta razón el Filtro de Kalman es conveniente cuando se necesita la estimación en tiempo real, debido a que puede ser implementado en forma muy eficiente.

El Filtro de Kalman modela tanto el estado a estimar como las variables asociadas a los sensores e información de los actuadores mediante variables aleatorias de distribución Gaussiana. Como en todo filtro Bayesiano, como parte del Filtro de Kalman se pueden distinguir claramente dos fases: la de *predicción* y la de *actualización*. Asociadas a ambas fases existen dos modelos probabilísticos: el *modelo de movimiento*, que describe la transición de un estado al siguiente, y el *modelo de observaciones*, que indica la relación entre una observación y el estado actual, lo que permite una corrección sobre este último.

### 2.3.2.1. Fase de predicción

La fase de predicción consiste en calcular el *prior belief* del estado, basándose en el estado anterior y la información del último movimiento realizado, aplicando el denominado *modelo de movimiento*. El estado obtenido como resultado de la fase de predicción en el tiempo  $t$  está representado entonces por una variable aleatoria de distribución normal con media  $\mathbf{x}_{t|t-1} \in \mathbb{R}^n$  y covarianza  $\mathbf{P}_{t|t-1} \in \mathbb{R}^{n \times n}$ .

La fase de predicción consiste entonces en calcular el estado  $\mathbf{x}_{t|t-1}$  y la covarianza  $\mathbf{P}_{t|t-1}$  de la siguiente manera:

$$\begin{aligned}\mathbf{x}_{t|t-1} &= \mathbf{F}_t \mathbf{x}_{t-1|t-1} + \mathbf{B}_t \mathbf{u}_t + \mathbf{w}_t \\ \mathbf{P}_{t|t-1} &= \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^T + \mathbf{Q}_t\end{aligned}$$

donde  $\mathbf{F}_t \in \mathbb{R}^{n \times n}$  y  $\mathbf{B}_t \in \mathbb{R}^{n \times n}$  conforman el modelo de movimiento,  $\mathbf{u}_t \in \mathbb{R}^n$  la información del movimiento realizado en el instante  $t$  y  $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$  es el error asociado al modelo, con  $\mathbf{Q}_t \in \mathbb{R}^{n \times n}$  la covarianza de dicho error.

### 2.3.2.2. Fase de actualización

La fase de actualización consiste en corregir la estimación obtenida en el paso anterior (predicción) a partir de incorporar las observaciones realizadas en el último instante. De esta forma, el Filtro de Kalman denota la media y covarianza del *posterior belief* en tiempo  $t$  como  $\mathbf{x}_{t|t} \in \mathbb{R}^n$  y  $\mathbf{P}_{t|t} \in \mathbb{R}^{n \times n}$  respectivamente.

En la fase de actualización se utiliza el modelo de observaciones  $\mathbf{H}_t \in \mathbb{R}^{n \times n}$  para vincular la medición  $\mathbf{z}_t \in \mathbb{R}^n$ , con el sensado esperado según el estado actual  $\mathbf{x}_{t|t-1} \in \mathbb{R}^n$  de la siguiente forma:

$$\mathbf{z}_t \simeq \mathbf{H}_t \mathbf{x}_{t|t-1} + \mathbf{v}_t$$

donde  $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}_t)$ , corresponde al ruido de medición y  $\mathbf{R}_t \in \mathbb{R}^{n \times n}$  es la matriz de covarianza asociada a dicho ruido.

Para corregir la estimación actual a partir de la nueva observación, se calcula primero:

$$\mathbf{y}_t = \mathbf{z}_t - \mathbf{H}_t \mathbf{x}_{t|t-1}$$

donde  $\mathbf{y}_t \in \mathbb{R}^n$  es la diferencia entre el sensado real  $\mathbf{z}_t$  y el estado predicho  $\mathbf{x}_{t|t-1}$ , transformado por el modelo de observaciones, lo que a su vez permite obtener el sensado que se esperaría obtener según dicho estado. Esta diferencia es conocida como innovación o medición residual.

Como siguiente paso se calcula la covarianza residual del siguiente modo:

$$\mathbf{S}_t = \mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \mathbf{R}_t$$

Luego, se obtiene  $\mathbf{K}_t \in \mathbb{R}^{n \times n}$ , la matriz de ganancia óptima que se utilizará para mejorar la estima-

ción:

$$\mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t^T \mathbf{S}_t^{-1}$$

Utilizando la ganancia  $\mathbf{K}_t$  y la diferencia calculada previamente  $\mathbf{y}_t$ , se actualiza la predicción  $\mathbf{x}_{t|t-1}$ , obteniendo la corrección de la estimación  $\mathbf{x}_{t|t}$  de la siguiente manera:

$$\mathbf{x}_{t|t} = \mathbf{x}_{t|t-1} + \mathbf{K}_t \mathbf{y}_t$$

Por último, se calcula la covarianza del estado corregido en tiempo  $t$  utilizando la matriz de ganancia  $\mathbf{K}_t$ , según:

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1}$$

Como resultado final del método se obtiene el estado estimado actualizado  $\mathbf{x}_{t|t}$  y su covarianza asociada actualizada  $\mathbf{P}_{t|t}$ .

### 2.3.3. Filtro de Kalman Extendido

Debido a que el Filtro de Kalman asume transiciones de estados lineales y un modelo de observaciones también lineal, esto suele representar una limitación en la práctica ya que para la mayor parte de los sistemas reales estas premisas no son válidas. Por esta razón, se propuso el filtro denominado Filtro de Kalman Extendido (*Extended Kalman Filter*) [20] que aborda dichas limitaciones.

De esta forma, se plantean las siguientes relaciones:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \mathbf{w}_{t-1}$$

$$\mathbf{z}_t = h(\mathbf{x}_t) + \mathbf{v}_t$$

en donde se introducen las funciones no-lineales  $f$  (modelo de movimiento) y  $h$  (modelo de observaciones). Como condición, ambas funciones deben ser diferenciables. Así para aplicar el Filtro de Kalman Extendido es necesario linealizar las funciones previamente mencionadas. Esto es realizable mediante una aproximación de Taylor, a partir del cálculo de los jacobianos  $\widehat{\mathbf{F}}$  y  $\widehat{\mathbf{H}}$  de la siguiente manera:

$$\widehat{\mathbf{F}} = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}_{t-1|t-1}, \mathbf{u}_{t-1}} \quad (2.2)$$

$$\widehat{\mathbf{H}} = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\mathbf{x}_{t|t-1}} \quad (2.3)$$

donde  $\widehat{\mathbf{F}}$  es el jacobiano de la función  $f$  evaluado en  $\mathbf{x}_{t-1|t-1}$  y  $\mathbf{u}_{t-1}$ , y  $\widehat{\mathbf{H}}$  es el jacobiano de la función  $h$  evaluado en  $\mathbf{x}_{t|t-1}$ .

Igual que en el algoritmo del Filtro de Kalman en este caso se identifican las mismas fases, la de

predicción y la de actualización. La primera queda definida de la siguiente manera:

$$\begin{aligned}\mathbf{x}_{t|t-1} &= f(\mathbf{x}_{t-1|t-1}, \mathbf{u}_{t-1}) \\ \mathbf{P}_{t|t-1} &= \widehat{\mathbf{F}} \mathbf{P}_{t-1|t-1} \widehat{\mathbf{F}}^T + \mathbf{Q}_{t-1}\end{aligned}\tag{2.4}$$

mientras que, la fase de actualización se define del siguiente modo:

$$\begin{aligned}\mathbf{y}_t &= \mathbf{z}_t - h(\mathbf{x}_{t|t-1}) \\ \mathbf{S}_t &= \widehat{\mathbf{H}} \mathbf{P}_{t|t-1} \widehat{\mathbf{H}}^T + \mathbf{R}_t \\ \mathbf{K}_t &= \mathbf{P}_{t|t-1} \widehat{\mathbf{H}}^T \mathbf{S}_t^{-1} \\ \mathbf{P}_{t|t} &= (\mathbf{I} - \mathbf{K}_t \widehat{\mathbf{H}}) \mathbf{P}_{t|t-1} \\ \mathbf{x}_{t|t} &= \mathbf{x}_{t|t-1} + \mathbf{K}_t \mathbf{y}_t\end{aligned}$$

La única diferencia en todos los pasos con respecto al algoritmo del Filtro de Kalman es la utilización de los jacobianos multiplicando covarianzas por las razones que se mencionaron anteriormente sobre la presunción de linealidad.

# Capítulo 3

## Método

En este capítulo se presenta el método de localización propuesto. Dado que este método está basado en el uso de múltiples cámaras, donde cada una localiza a un patrón de la escena utilizando el sistema WhyCon, el principal problema a abordar es la fusión de esta información.

Siendo que el sistema de localización WhyCon asociado a cada cámara estima la posición del patrón respecto de ésta, para resolver la fusión de todas las estimaciones el primer problema a abordar es el de llevar esta información a un único marco de referencia global. Para ello es necesario encontrar las transformaciones involucradas, que expresan la pose relativa entre las distintas cámaras y el marco de trabajo. Por esta razón, en este capítulo se presentan primero dos formas de encontrar las transformaciones correspondientes.

Teniendo entonces una forma de contar con diversas observaciones representadas en un único sistema, en este capítulo se presenta finalmente el método que permite fusionar esta información en una única estimación en forma robusta y precisa. Para ello, se propone el uso del Filtro de Kalman Extendido, el cual resuelve dicho problema a partir de considerar la incertidumbre asociada a cada observación individual y la del estimado actual.

### 3.1. Calibración multi-cámara

Al trabajar con múltiples cámaras que observan un mismo elemento del mundo, es necesario generalizar el modelo que se planteó en el capítulo anterior para pasar de un punto en el mundo 3D, a una imagen de la cámara en el plano 2D. En la figura 3.1 se ilustra el problema que se busca abordar.



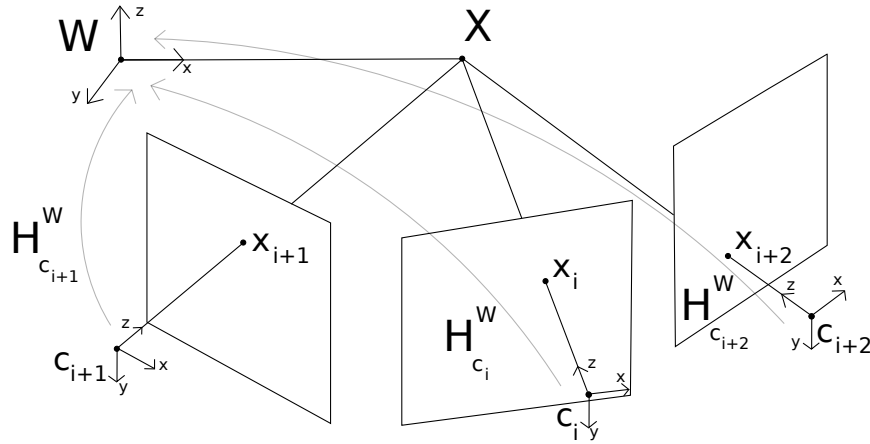


Figura 3.1: Modelo multicámara.

En la figura 3.1 se pueden ver tres cámaras  $C_i$  las cuales observan un punto  $\mathbf{x}_i$  en el plano de la imagen de cada una de ellas respectivamente. Cada cámara posee su propio sistema de coordenadas. El objetivo de calibrar las cámaras en este trabajo es obtener  $\mathbf{H}_{C_i}^W$  que permite realizar la transformación del punto  $\mathbf{X}$  que se encuentra en el sistema de coordenadas de la cámara al sistema de coordenadas del mundo  $W$  definido por el usuario. Para ello se plantean dos alternativas, calibrar las cámaras utilizando el sistema Kalibr, o calibrarlas utilizando el mismo sistema WhyCon.

### 3.1.1. WhyCon

Una opción para resolver la calibración multicámara es aprovechando la capacidad que tiene el sistema WhyCon para hallar la pose de la cámara respecto de un sistema de coordenadas dado. En este caso, dicho sistema supone que cada cámara ya se encuentra calibrada por separado (parámetros intrínsecos). Por lo tanto la calibración con este enfoque consiste en obtener los parámetros intrínsecos utilizando algún otro método y luego la transformación  $\mathbf{H}_{C_i}^W$  de cada cámara  $C_i$ , que permita trabajar respecto de un sistema  $W$  dado. Este proceso requiere definir el marco de referencia  $W$  a partir de colocar tres patrones definiendo los ejes  $X$  e  $Y$  sobre el plano de trabajo, y el origen del mismo como se observa en la figura 3.2.

El sistema WhyCon permite realizar la calibración mencionada para sistemas de coordenadas en 3D así como para sistemas de coordenadas en 2D (resolviendo la localización únicamente sobre el plano  $XY$ ).

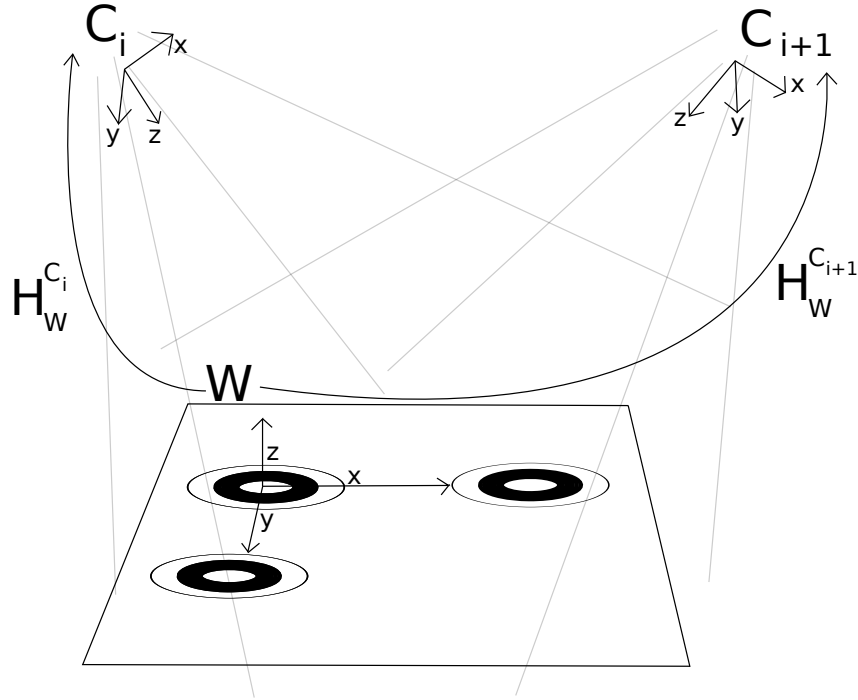


Figura 3.2: Sistema WhyCon con múltiples sensores.

### 3.1.1.1. Funcionamiento 3D

En el caso más general, donde se espera que el patrón se mueva en el espacio 3D, la transformación a encontrar para resolver el cambio de sistema de referencia consistirá de una traslación y rotación. Para hallar dicha transformación, WhyCon permite definir el marco de referencia a partir de colocar tres patrones, determinando los ejes  $X$  e  $Y$  sobre el plano de trabajo (generalmente el piso), y el origen del sistema (el eje  $Z$  quedará definido implícitamente a partir de los ejes previamente definidos).

La transformación que permite llevar un vector  $\mathbf{x}_c$  representado en referencia al sistema de coordenadas  $C$  de la cámara, a uno  $\mathbf{x}_w$  en referencia  $W$  del mundo, será:

$$\mathbf{x}_w = \mathbf{R} (\mathbf{x}_c - \mathbf{x}_0)$$

con:

$$\mathbf{R} = \left( \mathbf{t}_1^T \quad \mathbf{t}_2^T \quad \mathbf{t}_3^T \right)^{-1}$$

tal que:

$$\mathbf{t}_1 = \mathbf{x}_1 - \mathbf{x}_0$$

$$\mathbf{t}_2 = \mathbf{x}_2 - \mathbf{x}_0$$

$$\mathbf{t}_3 = \mathbf{t}_1 \times \mathbf{t}_2$$

donde  $\mathbf{x}_i$  son las coordenadas de los tres patrones utilizados para definir al sistema de referencia  $W$ .

### 3.1.1.2. Funcionamiento 2D

Existen otros casos de uso del sistema de localización en donde se sabe que los patrones, es decir los robots, se moverán en el plano. En este caso, es posible realizar una transformación de 3D en 2D, a un plano definido por el usuario.

La ventaja principal que provee trabajar en 2D es que se mejora la precisión con respecto a trabajar directamente en coordenadas 3D. Esto se debe a que, al realizar dicha transformación esencialmente se están transformando coordenadas 2D obtenidas en el plano de la imagen con coordenadas 2D del plano de trabajo. De esta forma, el error de estimación de la distancia al patrón no afecta la precisión de la estimación sino que esta depende únicamente del cálculo de la dirección del rayo de proyección del patrón hacia la cámara.

La transformación a utilizar en este caso, que permite corresponder posiciones 3D a posiciones 2D es una proyección. De esta forma, para hallarla, se requieren cuatro correspondencias conocidas para definirla. A partir de tener cuatro posiciones  $c_1, c_2, c_3, c_4$  conocidas que definen las esquinas de un cuadrado ubicado sobre el plano de trabajo, se puede obtener  $\mathbf{H}$  planteando las siguientes ecuaciones:

$$\mathbf{H} \hat{c}_1 = (0, 0, 1)$$

$$\mathbf{H} \hat{c}_2 = (1, 0, 1)$$

$$\mathbf{H} \hat{c}_3 = (0, 1, 1)$$

$$\mathbf{H} \hat{c}_4 = (1, 1, 1)$$

donde  $\mathbf{H}$  es una matriz de proyección de  $3 \times 3$  y donde  $\hat{c}_i = (c_i^x/c_i^z, c_i^y/c_i^z, 1)$ .

En este trabajo, de todas formas, se asumirá el caso más general de localización en 3D por lo que este modo de calibración no será utilizado.

### 3.1.2. Kalibr

Otra opción para obtener la calibración de las cámaras del sistema es utilizando Kalibr. En este caso dicho sistema permite obtener la transformación entre cada una de las cámaras del modelo (ver figura 3.3). Así, para resolver el problema propuesto originalmente en la figura 3.1, restaría encontrar una transformación entre alguna de estas cámaras y el sistema de coordenadas global  $W$ .

Kalibr en particular, permite obtener además de los parámetros extrínsecos, los parámetros

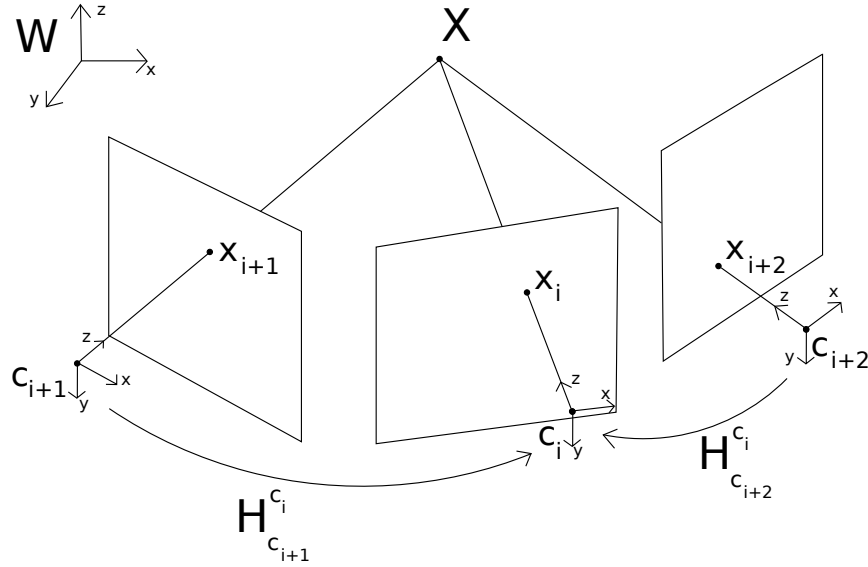


Figura 3.3: Modelo multicámara Kalibr.

intrínsecos de cada cámara. De esta manera, la calibración multi-cámara se reduce a encontrar:

$$\mathbf{x}_i = \mathbf{P}_i \mathbf{X}$$

en donde  $\mathbf{X}$  es un punto en el mundo 3D y  $\mathbf{P}_i$  es la matriz de proyección que permite pasar el punto del mundo 3D al plano de la imagen de la cámara  $C_i$ .

Como se mencionó en la sección 2.1, es posible representar a  $\mathbf{P}_i$  de la siguiente manera:

$$\mathbf{P}_i = \mathbf{K}_i [\mathbf{R}_i \mid \mathbf{t}_i]$$

con  $\mathbf{K}_i$  la matriz de parámetros intrínsecos y  $\mathbf{R}_i^i$ ,  $\mathbf{t}_i^i$  los parámetros extrínsecos de cada  $C_i$  que se buscan hallar.

Sin embargo, entre cada observación de las cámaras y la realidad existe un error, por ende no es  $\mathbf{P}_i$  lo que finalmente se obtiene sino  $\tilde{\mathbf{P}}_i$  donde:

$$\tilde{\mathbf{P}}_i \tilde{\mathbf{X}} \simeq \mathbf{P}_i \mathbf{X}$$

para cada cámara  $C_i$ .

Por lo tanto, para optimizar el resultado se buscan las proyecciones  $\tilde{\mathbf{P}}_i$  que minimicen ese error de la siguiente manera:

$$\min_{\tilde{\mathbf{P}}_i, \tilde{\mathbf{X}}} \sum_i d(\tilde{\mathbf{P}}_i \tilde{\mathbf{X}}, \mathbf{x}_i)^2 \quad (3.1)$$

donde  $d(a, b)$  es la distancia de la imagen geométrica entre los puntos homogéneos  $a$  y  $b$ . Esta estimación implica reducir al mínimo el error de reproyección y se conoce como *Bundle Adjustment* [32].

La calibración mediante Kalibr comprende esencialmente dos pasos. Primero, se adquiere una serie de imágenes con el sistema multi-cámara, moviendo un patrón especialmente diseñado (figura 3.4). Kalibr recupera de estas imágenes las correspondencias entre puntos de las imágenes  $\mathbf{x}_i$  y puntos  $\mathbf{X}$  del mundo (conociendo las dimensiones del patrón). Como segundo paso, a partir de estas correspondencias, se aplica el algoritmo de minimización que busca las matrices  $\mathbf{P}_i$ , según la función de error (3.1).

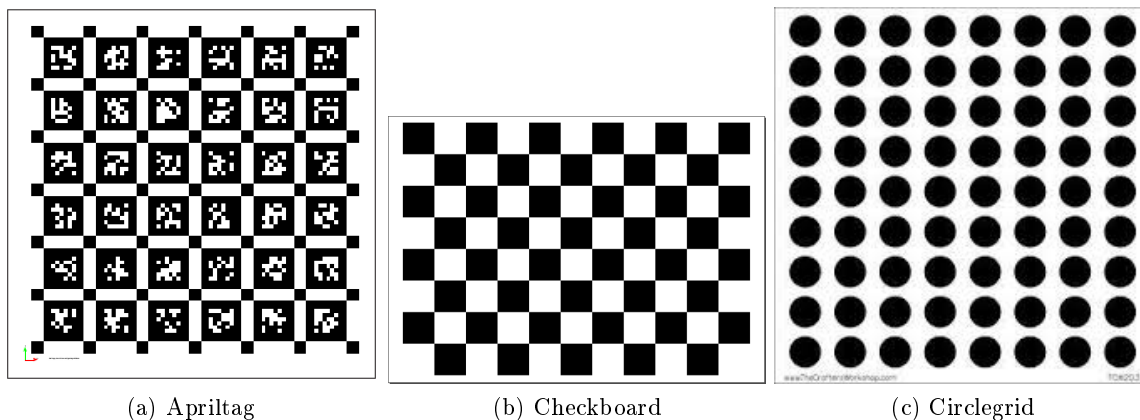


Figura 3.4: Patrones de calibración.

Como resultado de la calibración mediante Kalibr se obtienen las transformaciones  $\mathbf{H}_{C_{i+1}}^{C_i}$  para cada cámara  $C_i$ . Por lo tanto, para poder pasar del sistema de coordenadas de una cámara al sistema de coordenadas del mundo bastaría con hallar la transformación  $\mathbf{H}_{C_i}^W$  para cualquiera de las cámaras, ya que se podría transformar desde cualquier cámara a la cámara  $C_i$  y luego utilizar dicha transformación para finalmente trabajar en coordenadas del mundo. Dado que Kalibr no ofrece resolver este último problema, esto debería ser resuelto explícitamente a partir de, por ejemplo, colocar el patrón de calibración delante de alguna de las cámaras, determinando así el sistema de referencia.

## 3.2. Detalle del método

El método de localización propuesto en esta tesis permite localizar en forma precisa a un robot móvil utilizando un conjunto de cámaras. Para localizar al robot, el mismo lleva consigo un patrón visual que puede ser detectado por el sistema WhyCon (figura 2.3). De esta forma, cada cámara tendrá asociada una instancia del sistema WhyCon que procesa las imágenes capturadas y produce una estimación de la posición del patrón mientras se encuentre en su campo de visión. Con el fin de obtener una única estimación de la posición del patrón, el método propuesto utiliza el Filtro de Kalman Extendido, fusionando las distintas estimaciones asociadas a cada cámara.

Para utilizar el Filtro de Kalman Extendido es necesario definir previamente cual es el estado

que se quiere estimar y ambos modelos del filtro, como así también los jacobianos de los modelos.

En la figura 3.5 puede verse un esquema del sistema de localización propuesto. A partir de la observación  $\mathbf{z}_i$ , que describe la posición del patrón respecto del sistema de coordenadas asociado a la cámara  $C_i$ , se busca obtener una estimación  $\mathbf{x}_i$  que describa la posición del patrón respecto de un sistema global  $W$ . Además, se puede apreciar en dicha figura, la relación que existe entre los distintos sistemas de referencia (los de las cámaras y el de referencia, denotado como  $W$ ), dados por las transformaciones correspondientes. En este capítulo dichas transformaciones se asumen conocidas, habiendo sido halladas por alguna de las técnicas presentadas en las secciones anteriores.

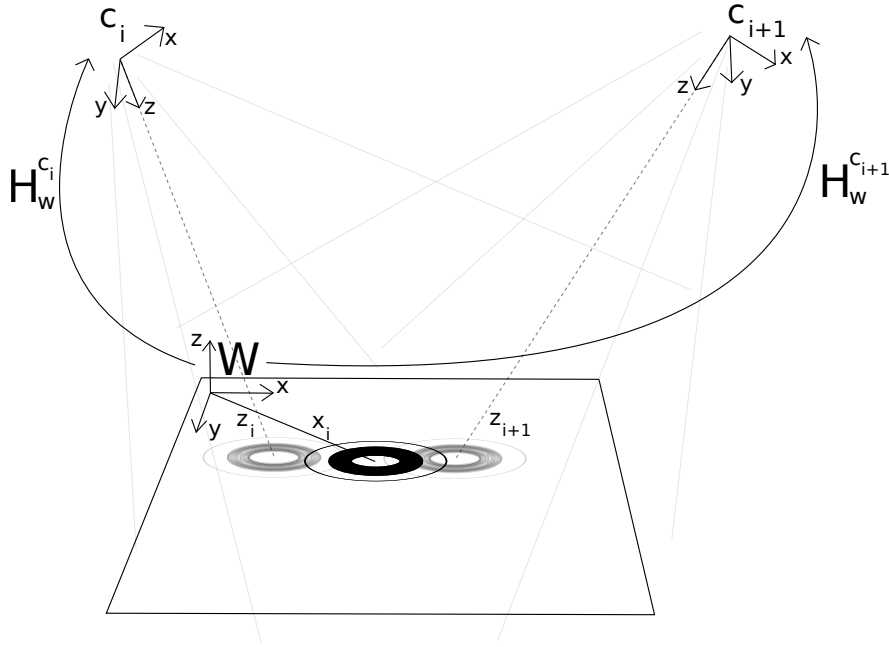


Figura 3.5: Modelo de Sensor.

Por otro lado, como se muestra en la figura 3.5, las observaciones individuales no necesariamente coinciden exactamente con la posición real del patrón debido a imprecisiones del sistema de sensado. Por esta razón es que el sistema de localización propuesto debe enfrentar el problema de integrar dos observaciones  $\mathbf{z}_i$  distintas, buscando acercarse lo más posible a la posición real del patrón. Para resolver este problema se propone entonces el uso del Filtro de Kalman Extendido. A continuación se plantea entonces la forma de instanciar el filtro  $EKF$  en el problema particular propuesto.

### 3.2.1. Estado

Como primer paso al utilizar el filtro  $EKF$  se debe definir el estado a estimar. Dado que en este trabajo se busca estimar la posición 3D del robot respecto de un sistema de coordenadas  $W$ , el estado estará representado por una variable aleatoria  $\mathbf{X}_t \sim \mathcal{N}(\mathbf{x}_t, \mathbf{P}_t)$ , de media  $\mathbf{x}_t \in \mathbb{R}^3$  y covarianza

$\mathbf{P}_t \in \mathbb{R}^{3 \times 3}$ , donde:

$$\mathbf{x}_t = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (3.2)$$

con  $x, y, z$  la posición en coordenadas 3D del patrón o robot a localizar.

Cabe aclarar que si bien el sistema de localización WhyCon permite localizar un patrón que se mueve en un plano con mayor precisión a uno que se mueve en todo el espacio 3D, justamente este segundo caso (más general) es el de interés en esta tesis y por ello se define al estado de la forma presentada.

### 3.2.2. Modelo de movimiento

Como parte del filtro *EKF*, el modelo de movimiento es el modelo que permite predecir la nueva posición del patrón en el instante siguiente. Para definir dicho modelo, se debe también definir la entrada  $\mathbf{u}_t$  del mismo. Si se define  $\mathbf{u}_t = \mathbf{v}_t$ , donde  $\mathbf{v}_t$  corresponde a una estimación de la velocidad lineal del patrón, se puede plantear el siguiente modelo de movimiento:

$$f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}_{t-1}) = \begin{pmatrix} x_{t-1} + \Delta t u_x + w_x \\ y_{t-1} + \Delta t u_y + w_y \\ z_{t-1} + \Delta t u_z + w_z \end{pmatrix} \quad (3.3)$$

donde  $\mathbf{w}$  corresponde al ruido asociado al modelo de movimiento, con  $\mathbf{w} \sim \mathcal{N}(0, \mathbf{Q})$ .

Con respecto a  $\mathbf{v}_t$ , esta puede ser obtenida directamente a partir de un sensor propio del robot (por ejemplo, de la odometría obtenida mediante encoders) o partir de tomar la diferencia de posiciones de las observaciones (estos dos enfoques son explicados en las secciones 3.2.2.1 y 3.2.2.2).

Cabe aclarar que aunque en este caso  $f$  sea lineal, podría plantearse un modelo no-lineal (por ejemplo, si se tuviera en cuenta la orientación). Por lo tanto, en esta tesis se asume igualmente el caso en el cual la función del modelo es no lineal, obteniendo el método de forma general, para luego instanciarlo en nuestro problema particular. De esta manera, para linealizar  $f$  y poder utilizar el Filtro de Kalman Extendido es necesario calcular los jacobianos de dicha función (ver ecuación (2.2)). Por ende, se deben hallar las derivadas parciales de  $f$  con respecto al vector  $\mathbf{x}$  y con respecto al vector  $\mathbf{w}$  dando lugar a los jacobianos  $\mathbf{A}$  y  $\mathbf{W}$ , respectivamente:

$$\mathbf{A} = \frac{\delta f}{\delta \mathbf{x}} \quad \mathbf{W} = \frac{\delta f}{\delta \mathbf{w}}$$

A partir de la ecuación (3.2) y (3.3), los jacobianos quedan entonces definidos como:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Por otra parte, se debe considerar que las entradas del modelo de movimiento poseen su propia incertidumbre, en nuestro caso la velocidad tiene asociado un ruido aditivo  $\xi_{\mathbf{u}}$ . Asumiendo que este ruido puede ser modelado como una variable gaussiana de media 0 y covarianza  $\mathbf{P}_{\mathbf{u}}$ , entonces se puede proponer el siguiente modelo de movimiento:

$$f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1} + \xi_{\mathbf{u}_{t-1}}, \mathbf{w}_{t-1}) = \begin{pmatrix} x_{t-1} + \Delta t (u_x + \xi_{u_x}) + w_x \\ y_{t-1} + \Delta t (u_y + \xi_{u_y}) + w_y \\ z_{t-1} + \Delta t (u_z + \xi_{u_z}) + w_z \end{pmatrix} \quad (3.4)$$

Como parte del proceso de linealización requerido por el filtro *EKF*, se debe entonces calcular el jacobiano  $\mathbf{J}_{\mathbf{u}}$  de  $f$  con respecto a  $\mathbf{u}$ :

$$\mathbf{J}_{\mathbf{u}} = \frac{\delta f}{\delta \mathbf{u}}$$

que en base a (3.4) queda definido como:

$$\mathbf{J}_{\mathbf{u}} = \begin{bmatrix} \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix}$$

Con este jacobiano entonces puede propagarse esta covarianza para obtener la contribución a la covarianza total asociada al cálculo de la covarianza resultante tras realizar el paso de predicción:

$$\mathbf{Q}_{\mathbf{u}} = \mathbf{J}_{\mathbf{u}} \mathbf{P}_{\mathbf{u}} \mathbf{J}_{\mathbf{u}}^T$$

De esta forma, la matriz de covarianza resultante queda dada por:

$$\mathbf{Q}' = \mathbf{Q} + \mathbf{Q}_{\mathbf{u}}$$

donde  $\mathbf{Q}'$  será la matriz a utilizar en el paso de predicción de *EKF* (2.4).

### 3.2.2.1. Velocidad obtenida por odometría

Para obtener un estimado de la velocidad a la que se mueve el patrón a seguir, se puede tomar directamente la velocidad obtenida mediante sensores internos del robot. Por ejemplo, para un robot terrestre pueden utilizarse los encoders asociados a las ruedas. Esta estimación de velocidad



es suficientemente precisa a corto plazo como para proveer una buena predicción del movimiento del patrón. En este caso, esta velocidad estimada es utilizada directamente como entrada al modelo de predicción.

### 3.2.2.2. Velocidad estimada a partir de observaciones

En caso de que no se pueda asumir la existencia de un sensor que provea información de velocidad, se desea poder estimar ésta de alguna otra forma. En este sentido es que se propone como alternativa utilizar directamente las diferencias de las posiciones observadas por las cámaras para estimar la velocidad de movimiento del patrón. Debido a que dicha diferencia sería demasiado ruidosa, se plantea un filtrado simple sobre esta variable. Concretamente, se propone utilizar un filtro pasa bajos, que se puede definir de la siguiente forma:

$$\mathbf{v}_t = \alpha (\mathbf{x}_t^{obs} - \mathbf{x}_{t-1}^{obs}) / \Delta t^{obs} + (1 - \alpha) \mathbf{v}_{t-1}$$

donde  $\alpha$  es el coeficiente del filtro,  $\mathbf{x}^{obs}$  es la posición observada por los sensores y  $\Delta t^{obs}$  es el tiempo entre la observación actual y la anterior.

Debido a que se tienen diversas cámaras produciendo distintas estimaciones, la velocidad es actualizada utilizando pares de observaciones de una misma cámara. Asimismo, la velocidad es actualizada con este filtro cada vez que se tenga una nueva observación.

La ventaja de utilizar este enfoque es que simplifica el uso del sistema, haciendo del robot un objeto pasivo que se sigue mediante observaciones externas, sin incluir el sistema de sensado del robot como parte del mismo. Sin embargo, como contrapartida, se debe aclarar que no se está considerando un modelo de error como el utilizado como parte del filtro *EKF*. De esta forma, se espera que con dicho enfoque posiblemente en ciertas situaciones el estimado de velocidad sufra de errores asociados al sistema de sensado utilizado. Aún así, resulta interesante analizar el desempeño del sistema al considerar el caso en el que no se cuente con odometría. Por esta razón, en adelante se utilizará esta forma de estimación de la velocidad.

### 3.2.3. Modelo de observaciones

El modelo de observaciones utiliza la información de los sensores para mejorar la precisión de la estimación del estado durante el paso de corrección del filtro *EKF*. Asumiendo que cada una de las instancias del sistema WhyCon arrojarán observaciones sobre la posición del patrón respecto del sistema de referencia  $W$ , el modelo de observación se reduce a:

$$h(\mathbf{x}_{t-1}, \mathbf{v}_{t-1}) = \mathbf{x}_{t-1} + \mathbf{v}_{t-1} \quad (3.5)$$

donde  $\mathbf{v} \sim \mathcal{N}(0, \mathbf{R})$  corresponde al ruido de observación.

Se puede observar en la ecuación (3.5) que  $h$  depende directamente de  $\mathbf{x}_{t-1}$ . Esto se debe a que se utiliza directamente el valor obtenido por la observación para actualizar la estimación. Tal como ocurre en el modelo de movimiento, para linealizar  $h$  y poder utilizar el filtro *EKF* es necesario calcular los jacobianos de dicha función (ver ecuación (2.3)). Por lo tanto, se deben hallar las derivadas parciales de  $h$  con respecto al vector  $\mathbf{x}$  y con respecto al vector  $\mathbf{w}$  dando lugar a los jacobianos  $\mathbf{H}$  y  $\mathbf{V}$  respectivamente:

$$\mathbf{H} = \frac{\delta h}{\delta \mathbf{x}} \quad \mathbf{V} = \frac{\delta h}{\delta \mathbf{v}}$$

A partir de la ecuación (3.2) y (3.5), los jacobianos quedan entonces definidos como:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{V} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Cabe aclarar que la simplicidad de este modelo de observación radica en el uso de un único sistema de coordenadas  $W$  por lo que la observación esperada (dada por el modelo de observación) equivale directamente al estimado que se tiene actualmente. Para utilizar el filtro *EKF* así definido se deben resolver entonces dos cuestiones al ejecutar el paso de corrección ante un sensado  $\mathbf{z}_i$ : transformar la observación  $\mathbf{z}_i$  de  $C_i$  a  $W$  y obtener la matriz de covarianza  $\mathbf{R}$  asociada a  $\mathbf{z}_i$ , también representada respecto de  $W$ .

Por lo tanto, se busca la covarianza  $\mathbf{R}$  que represente el modelo de error del sistema WhyCon. Dicho modelo dependerá directamente a la distancia entre el patrón y la cámara y a los ángulos de la misma. En este trabajo los errores angulares,  $e_\alpha$  y  $e_\beta$ , son considerados constantes y calculados en base a la distancia focal, al campo de visión de la cámara y al error en píxeles del sistema WhyCon. Por otro lado, el error en distancia es planteado en [16] del siguiente modo:

$$e_\rho = \frac{\Delta f}{f_x} + \Delta e_0 \frac{\mathbf{x}f_x}{d_0} + \eta_{rad}$$

con  $\Delta f$  el error de la estimación de la distancia focal,  $\Delta e$  el error de la estimación de los ejes de la elipse,  $d_0$  el diámetro del círculo negro del patrón,  $\eta_{rad}$  el error relativo de la distorsión radial del modelo. Las variables  $\Delta f$  y  $\eta_{rad}$  pueden ser calculados por la calibración de la cámara, pero  $\Delta e_0$  es influido por muchos factores, como por ejemplo la luz o el ruido termal de la cámara. En particular, en los experimentos de [16],  $\Delta e_0$  tiene un promedio de 0,15 píxeles. Sin embargo, en el caso general, estos parámetros deben ser encontrados empíricamente.

Por ende, la covarianza asociada a una observación  $\mathbf{z}_i$ , que se asume en coordenadas polares, será de la forma:

$$\mathbf{R}' = \begin{bmatrix} e_\rho & 0 & 0 \\ 0 & e_\alpha & 0 \\ 0 & 0 & e_\beta \end{bmatrix}$$

Finalmente para obtener el modelo deseado es necesario propagar  $\mathbf{R}'$  a una covarianza en coor-

denadas cartesianas y respecto del sistema  $W$ . Para ello, dada una observación  $\hat{\mathbf{z}}_i$  se define la función  $h(\hat{\mathbf{z}}_i)$  de esta forma:

$$h(\hat{\mathbf{z}}_i) = \mathbf{H}_{C_i}^W PaC(\hat{\mathbf{z}}_i)$$

donde  $\mathbf{H}_{C_i}^W$  es la transformación del sistema de coordenadas de la cámara  $C_i$  al sistema  $W$  y  $PaC$  la función para convertir de coordenadas polares a cartesianas definida del siguiente modo:

$$PaC(\rho, \alpha, \beta) = (\rho \cos(\alpha) \cos(\beta), \rho \cos(\alpha) \sin(\beta), \rho \sin(\alpha))$$

Luego, para propagar  $\mathbf{R}'$  se debe hallar el jacobiano de la función  $h$  con respecto a  $\hat{\mathbf{z}}_i$ :

$$\mathbf{J}_h = \frac{\partial h}{\partial \hat{\mathbf{z}}_i}$$

donde por regla de la cadena resulta:

$$\mathbf{J}_h = \mathbf{R}_i \mathbf{J}_{PaC}(\hat{\mathbf{z}}_i)$$

con  $\mathbf{R}_i$  la matriz de rotación perteneciente a la transformación  $\mathbf{H}_{C_i}^W$  y el jacobiano de la función  $PaC$  evaluado en la observación  $\hat{\mathbf{z}}_i$  queda definido de la siguiente manera:

$$\mathbf{J}_{PaC} = \begin{bmatrix} \cos(\alpha) \cos(\beta) & -\rho \sin(\alpha) \cos(\beta) & -\rho \cos(\alpha) \sin(\beta) \\ \cos(\alpha) \sin(\beta) & -\rho \sin(\alpha) \sin(\beta) & \rho \cos(\alpha) \cos(\beta) \\ \sin(\alpha) & \rho \cos(\alpha) & 0 \end{bmatrix}$$

Por último queda definida  $\mathbf{R}$  la matriz de covarianza propagada del modelo del siguiente modo:

$$\mathbf{R} = \mathbf{J}_h \mathbf{R}' \mathbf{J}_h^T$$

## Capítulo 4

# Experimentos

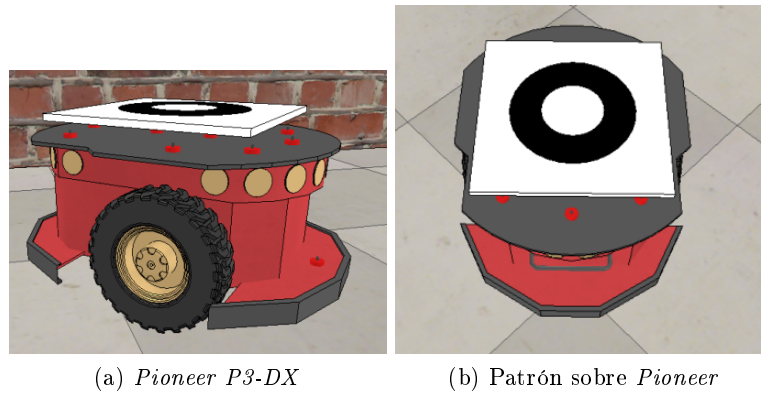
En este capítulo se presentan los resultados obtenidos a partir de realizar una serie de experimentos utilizando el método de localización propuesto en esta tesis.

En primer lugar, se presentan los experimentos relacionados a la calibración de las cámaras. El primer sistema utilizado para obtener la calibración es Kalibr [33] detallado en la sección 3.1.2. El otro sistema utilizado para calibrar las cámaras es el mismo WhyCon [16], método que se detalló en la sección 3.1.1. Con las cámaras calibradas se obtiene la posición relativa de las mismas.

Finalmente, se detalla como se definió el modelo de error en función de la distancia y el valor del parámetro del filtro de la velocidad.

En una segunda etapa, se pone a prueba el funcionamiento del método sobre diferentes escenarios, los mismos fueron definidos con el fin de evaluar la precisión del sistema en diferentes condiciones. Como primer experimento se desea analizar la precisión del método cuando las cámaras poseen máximo solapamiento, es decir, que las dos cámaras vean al patrón a lo largo de todo el recorrido. El siguiente experimento consiste en evaluar la precisión al agregar más cámaras que vean la escena. Por otra parte, se desea evaluar la precisión del método cuando el solapamiento es mínimo, es decir cuando se busca maximizar el área de cobertura. Para finalizar con las pruebas, se realizó un experimento general en donde se aplica el conocimiento del desempeño del sistema, obtenido mediante los experimentos previos, en un caso que se asemeje a una situación real.

Con el fin de evaluar con precisión el desempeño del método y de poder evaluar fácilmente distintas configuraciones y escenarios de prueba sin requerir una infraestructura compleja para los experimentos (sistema de cámaras interconectadas), en todos los experimentos se trabajó en simulación. La herramienta de simulación que se utilizó es V-Rep (*Virtual Robot Experimentation Platform*) [34], con la cual se diseñaron diversos escenarios acorde a cada experimento. En todas las simulaciones se cuenta con un robot *Pioneer 3-DX*, figura 4.1, el cual posee un patrón circular adosado a su chasis.

Figura 4.1: *Pioneer P3-DX*.

El robot fue controlado manualmente durante los experimentos, mediante comandos de velocidad. Los escenarios cuentan con cámaras ubicadas en distintas posiciones. Las imágenes obtenidas son procesadas por instancias individuales del sistema WhyCon. Los resultados de localización de dichas instancias son tomados como entrada al sistema de localización propuesto.

El simulador permite conocer además las posiciones reales de los elementos de la escena (*ground truth*), en particular la del patrón, esta información fue utilizada para contrastar con los resultados obtenidos por el sistema de localización propuesto. Las medidas de dicho patrón son:  $0,2\text{ m}$  para el diámetro externo (círculo negro) y  $0,1\text{ m}$  para el diámetro interno (círculo blanco). Las cámaras que se simularon se configuraron con resolución de  $800\text{ px} \times 600\text{ px}$ , un ángulo de visión de  $60^\circ$  y longitud focal de  $682\text{ px}$ . Esta configuración corresponde a una situación típica usada comúnmente con el sistema de localización WhyCon.

## 4.1. Calibración

Como se explicó anteriormente en la sección 3.1, se cuenta con dos formas de realizar la calibración multicámara, utilizando Kalibr o el mismo WhyCon. El objetivo de los siguientes experimentos es evaluar ambos casos para determinar la conveniencia en base a los resultados obtenidos de utilizar un sistema u el otro en una situación real.

En este trabajo en particular, siendo que en los experimentos posteriores se trabajó en simulación, una tercera opción es utilizar directamente la información que da V-Rep respecto de la pose de cada cámara de la escena. Por esta razón, de aquí en adelante, la calibración se supondrá conocida y no se procederá a realizar la calibración explícitamente.

#### 4.1.1. Kalibr

Para evaluar el sistema Kalibr se realizaron dos experimentos que variaron en cuanto a la pose relativa de las cámaras. En este experimento se utilizaron dos cámaras USB modelo *Firefly Mv* de  $640\text{ px} \times 480\text{ px}$ .

En el primer caso se colocaron las cámaras en configuración estéreo (es decir, perfectamente paralelas), con *baseline* de  $0,14\text{ m}$ , mientras que en el segundo una cámara se posicionó a  $45^\circ$  respecto de la otra, en donde el *baseline* fue de  $1,80\text{ m}$ .

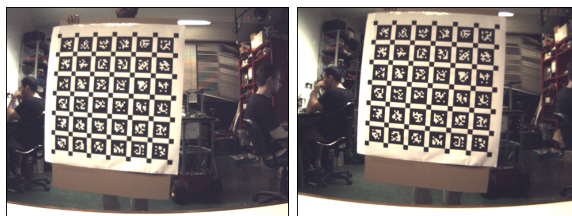
Para realizar la calibración se utilizó un tablero conteniendo una grilla de marcadores del sistema Kalibr y se tomaron imágenes con ambas cámaras variando la posición del mismo. En este paso lo que se busca es cubrir todas las posibles áreas que visualiza cada cámara en la escena, lejos, cerca, en un costado, en otro, inclinado hacia arriba o hacia abajo, etc.. Algunos de estos movimientos registrados se muestran en la figura 4.2 para el experimento con las cámaras estéreo y en la figura 4.3 para el experimento con una cámara a  $45^\circ$ . Se puede ver como en el segundo caso, la posición relativa de las cámaras genera una diferencia importante en la perspectiva con la cual se observa el tablero de calibración.

Luego de ejecutar el sistema Kalibr con las imágenes adquiridas, se obtuvieron los siguientes resultados: en la figura 4.4 se presentan los resultados de la calibración (posición relativa de cada cámara) y en la figura 4.5, los resultados de la detección del tablero en todas las imágenes junto al error de reproyección de los patrones que lo componen.

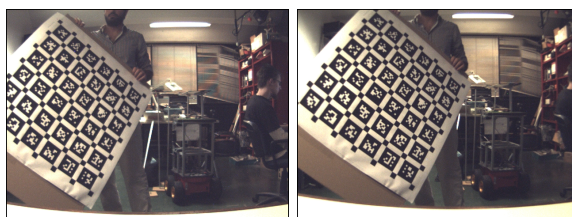
Para el caso con las cámaras estéreo, el *baseline* que se obtuvo como resultado de la calibración es de  $0,136\text{ m}$ , y los ángulos de una cámara con respecto a la otra son, para el eje  $X = 0,624^\circ$ , para el eje  $Y = -1,753^\circ$  y para el eje  $Z = -0,572^\circ$ . Por otra parte, para el caso con las cámaras a  $45^\circ$ , el *baseline* resultante es de  $1,76\text{ m}$ , y los ángulos de una cámara con respecto a la otra son, para el eje  $X = -2,452^\circ$ , para el eje  $Y = 51,310^\circ$  (el de rotación de la cámara) y para el eje  $Z = 4,498^\circ$ .

A partir de los experimentos realizados puede verse que, en principio, el sistema Kalibr parece obtener correctamente la posición relativa de las cámaras. De todas formas, estos experimentos representan mayormente una prueba de concepto y se requeriría realizar posteriores experimentos en diversas configuraciones para asegurarse de la precisión que puede obtenerse.

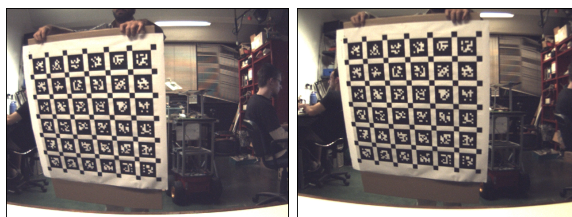
Por último, como se mencionó en la sección 3.1.2, para completar el experimento, es necesario calcular la pose de una de las cámaras con respecto a un sistema de coordenadas externo. Esto también queda pendiente para futuros experimentos.



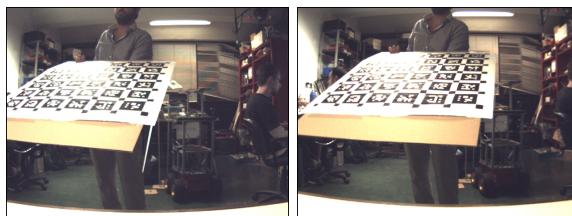
(a) camera0: patrón de frente. (b) camera1: patrón de frente.



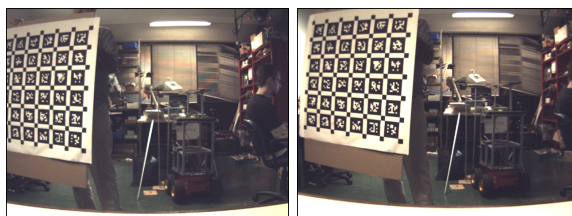
(c) camera0: patrón girado sobre su eje. (d) camera1: patrón girado sobre su eje.



(e) camera0: patrón inclinado hacia abajo. (f) camera1: patrón inclinado hacia abajo.

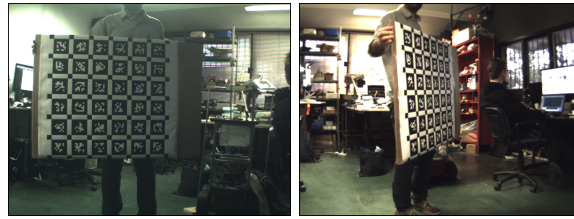


(g) camera0: patrón inclinado hacia arriba. (h) camera1: patrón inclinado hacia arriba.

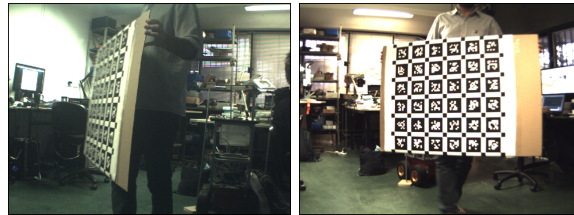


(i) camera0: esquina superior izquierda. (j) camera1: esquina superior izquierda.

Figura 4.2: Movimientos con el patrón utilizando cámaras estéreo. En la columna izquierda se muestra la visión de camera0 y en la derecha camera1.



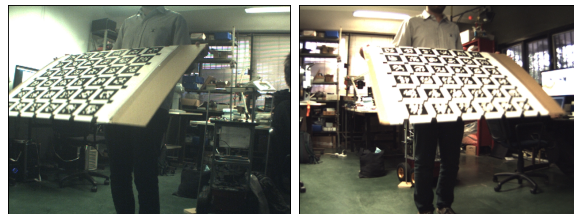
(a) camera0: patrón hacia camera0 de frente. (b) camera1: patrón hacia camera 0 de frente.



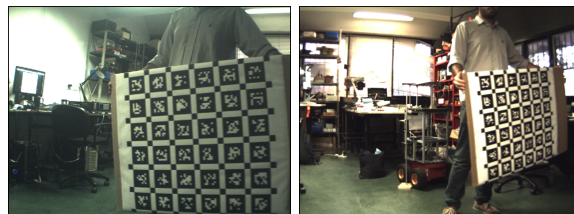
(c) camera0: patrón hacia camera1 de frente. (d) camera1: patrón hacia camera1 de frente.



(e) camera0: patrón inclinado hacia abajo. (f) camera1: patrón inclinado hacia abajo.



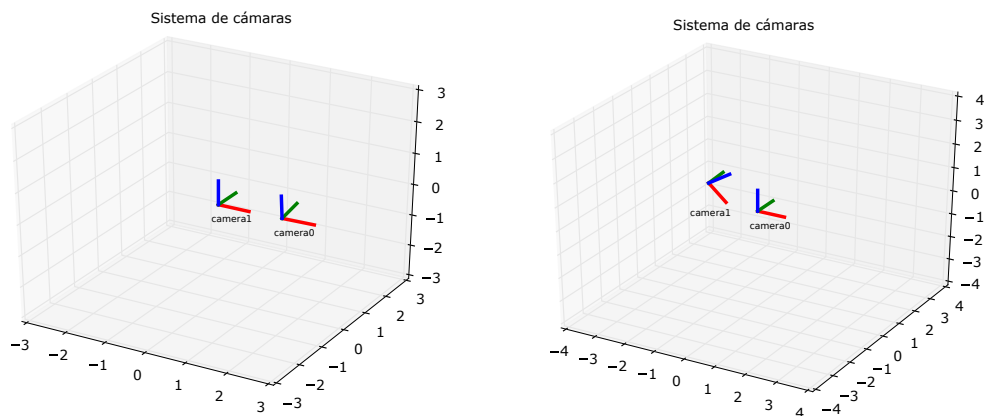
(g) camera0: patrón inclinado hacia arriba (h) camera1: patrón inclinado hacia arriba.



(i) camera0: esquina inferior derecha (j) camera1: esquina inferior derecha.

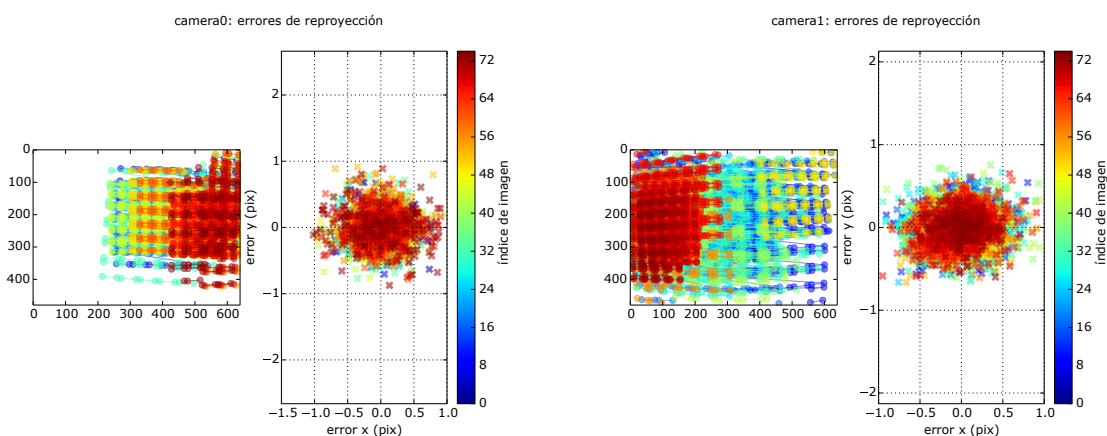
Figura 4.3: Movimientos con el patrón utilizando cámaras a  $45^\circ$ . En la columna izquierda se muestra la visión de camera0 y en la derecha camera1.



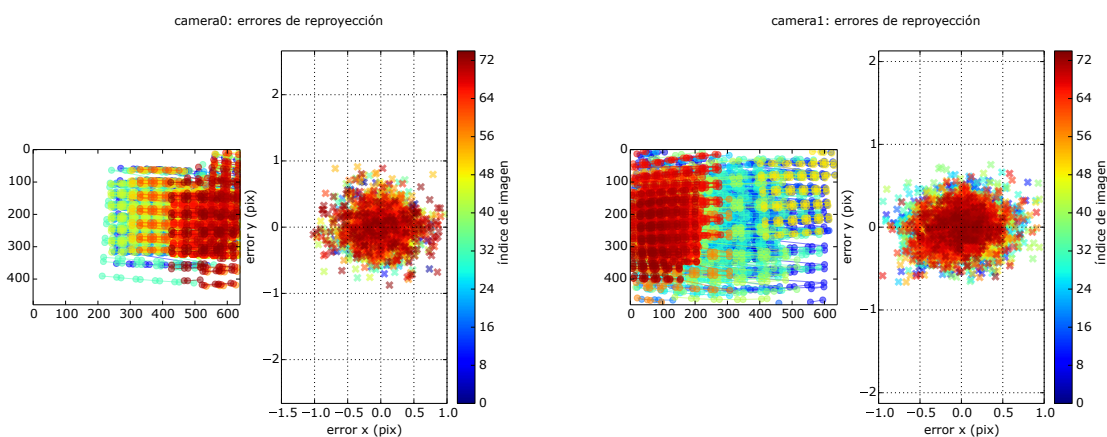


(a) Posición estimada de las cámaras estéreo. (b) Posición estimada de las cámaras a 45°.

Figura 4.4: Estimación de la posición de las cámaras.



(a) Caso estéreo, errores de reproyección para camera0. (b) Caso estéreo, errores de reproyección para camera1.



(c) Caso 45°, errores de reproyección para camera0. (d) Caso 45°, errores de reproyección para camera1.

Figura 4.5: Errores de reproyección en píxeles. En cada figura a la izquierda se muestran las detecciones del tablero y a la derecha los errores de reproyección.

### 4.1.2. WhyCon

Continuando con el proceso de calibración multicámara, en esta sección se desea evaluar el funcionamiento del sistema WhyCon para resolver dicho proceso. En este caso, a diferencia de los experimentos realizados con Kalibr, los experimentos se realizaron en el simulador V-Rep, por cuestiones de simplicidad. La figura 4.6 muestra la escena utilizada para poder calibrar una cámara que observa los patrones en el piso de la habitación.

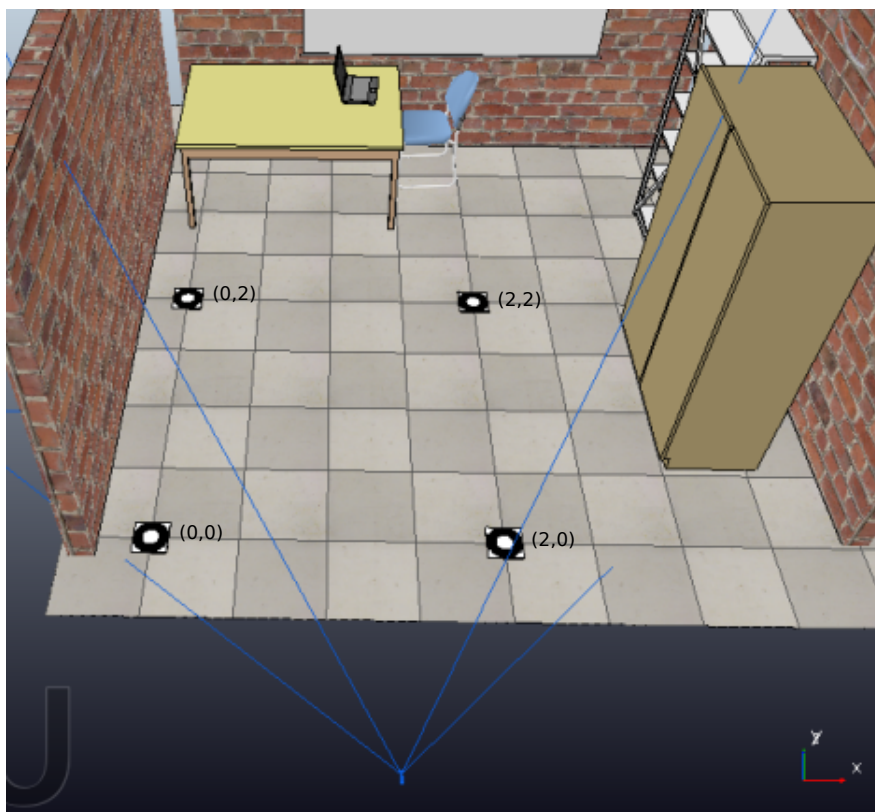


Figura 4.6: Escena con una cámara y los patrones para calibrar con WhyCon.

En la escena se observan cuatro patrones porque así lo requiere el sistema, que obtiene una calibración 2D y 3D simultáneamente. De todas formas, para la calibración 3D solo se utilizan tres de los mismos, como se mencionó en la sección 3.1.1.

La calibración obtenida mediante WhyCon provee la pose de cada cámara con respecto al origen del sistema de coordenadas del mundo definido por dichos patrones. Por esta razón se colocó el patrón que determina el origen del sistema de forma que coincidiera con el origen del sistema de coordenadas que utiliza el simulador. De esta forma, se pudo comparar la pose obtenida para cada cámara con la real.

Se pueden observar los resultados obtenidos por la calibración utilizando el sistema WhyCon en comparación con los valores reales (*ground truth*) en la tabla 4.1.

	Posición [m]			Ángulos [°]		
	$x$	$y$	$z$	$\alpha$	$\beta$	$\gamma$
<i>Ground truth</i>	1,225	-2	1,755	-110	0	0
WhyCon	1,225	-2,014	1,747	-109,839	0,121	-0,085

Cuadro 4.1: Comparación de la calibración utilizando WhyCon. Donde  $(x, y, z)$  representa la posición de la cámara en metros y  $(\alpha, \beta, \gamma)$  los ángulos de rotación en grados sobre los ejes respectivamente.

A partir de los resultados obtenidos se puede ver que la pose estimada mediante el sistema WhyCon es muy cercana a la real, concluyendo que este método es una alternativa válida para afrontar el problema de la calibración. Sin embargo, como trabajo futuro se plantea comparar la precisión de calibración utilizando el método WhyCon y el método Kalibr. Como se mencionó en la sección 3.1.2, para poder comparar estos dos métodos, hace falta primero calcular la pose de una de las cámaras con respecto a un sistema de coordenadas externo, dado que Kalibr no lo provee.

## 4.2. Modelo de error en función de la distancia

El objetivo de este experimento es obtener un modelo de error para el sistema WhyCon, en función de la distancia del patrón a la cámara y de los ángulos del rayo de proyección, que une el patrón con esta última. En cuanto a la distancia, se parte del modelo planteado en [16] y se buscan obtener experimentalmente los parámetros desconocidos. Por otro lado, se consideraron modelos alternativos (no-lineales), buscando un mejor ajuste a los datos obtenidos experimentalmente.

La razón de buscar dicho modelo de error es que esto permite incorporar esta información al filtro *EKF* mediante la matriz de covarianza asociada a una observación. En otras palabras, se construirá una matriz de covarianza donde la varianza sea proporcional al error esperado en cada dimensión. Como se mencionó en la sección 3.2.3, lo que se busca entonces es definir una matriz de covarianza para una observación representada en coordenadas polares, relativas a la cámara desde la cual se obtuvo. A partir de dicha matriz, se puede propagar la misma a una covarianza asociada a una medición absoluta (respecto del sistema global  $W$ ) y en coordenadas cartesianas, que es lo que espera el filtro de *EKF* para una observación. En resumen, lo que se busca es obtener un error de estimación para la distancia al patrón y los ángulos del rayo de proyección.

En primer lugar, en cuanto al error angular, se sabe que el sistema WhyCon determina la posición  $(x, y)$  del patrón con una precisión *sub-píxel*, por lo que se toma en forma genérica un valor de error de  $0,5px$ . De esta forma, teniendo un valor en píxeles, para obtener los valores  $e_\alpha$  y  $e_\beta$ , se debe establecer una relación entre el error en píxeles de WhyCon con la el error angular que debe ser medido en radianes. Esta relación se obtiene del siguiente modo:

$$e_\alpha = \frac{p \times a}{f} \frac{\pi}{180}$$

donde  $p$  es el error en píxeles,  $a$  el *field of view* y  $f$  la longitud focal.

Teniendo que  $f_x = 800 px$ ,  $f_y = 600 px$  y  $a = 60^\circ$ , se obtiene entonces un error angular de  $0,0875 rad$  y  $0,005 rad$  para  $e_\alpha$  y  $e_\beta$  respectivamente.

En segundo lugar, respecto del error en función de la distancia, para hallar los coeficientes del modelo de error (y evaluar modelos alternativos), se realizó un experimento que permitió establecer la relación entre la distancia del patrón a la cámara y el error de localización cometido. Utilizando el simulador V-Rep, se diseñó la escena que se muestra en 4.7, donde se muestra el movimiento del robot durante el experimento, que se fue alejando progresivamente de la cámara, yendo de una punta de la habitación hacia la otra (ver esquema también en figura 4.8).

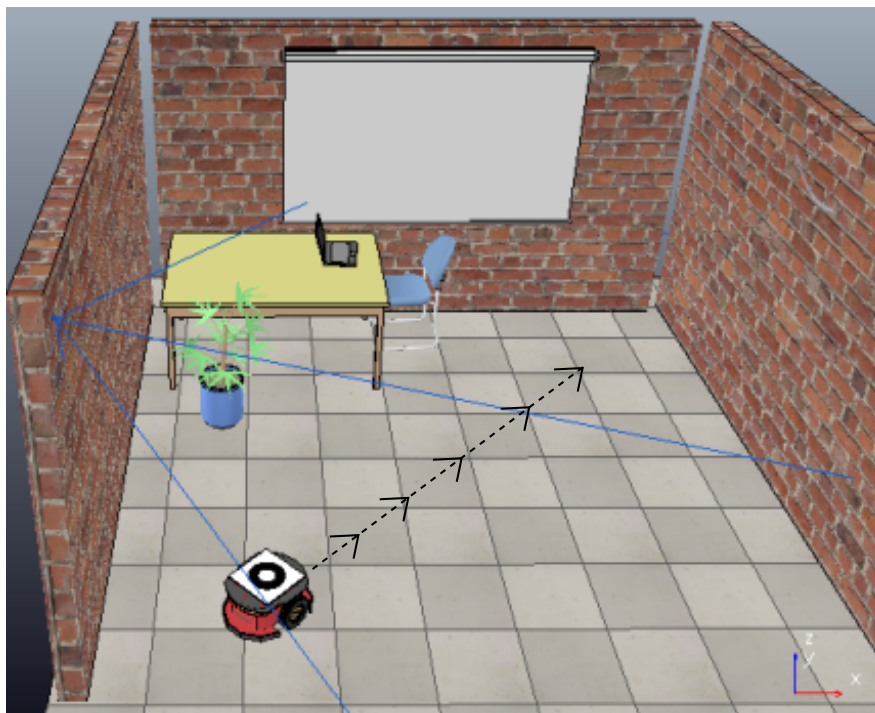


Figura 4.7: Escena con una cámara siguiendo el recorrido de un robot Pioneer el cual traslada el patrón encima de él paralelo al piso.

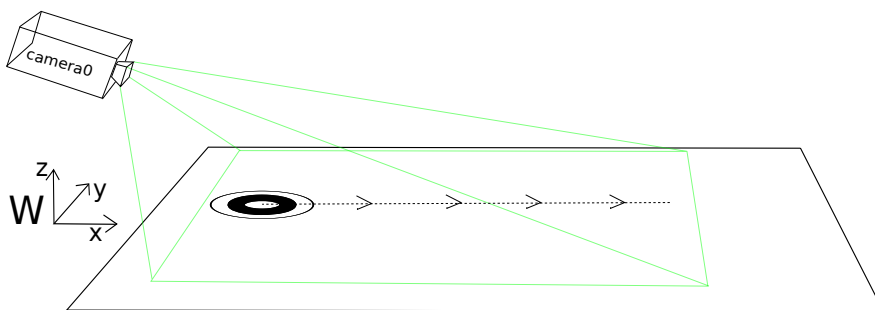


Figura 4.8: Esquema del recorrido en la escena con una cámara.

Comparando los valores de la posición real del patrón (*ground truth*) con la estimada por Why-Con, se calcula la norma del error en cada instante registrando a qué distancia se encuentra el patrón a cada momento de la cámara. Los valores obtenidos fueron utilizados para estimar la relación entre distancia y norma del error, ajustando con una función lineal (como la propuesta en [16]) y, además, con una función cuadrática y una exponencial. Los resultados se presentan en la figura 4.9.

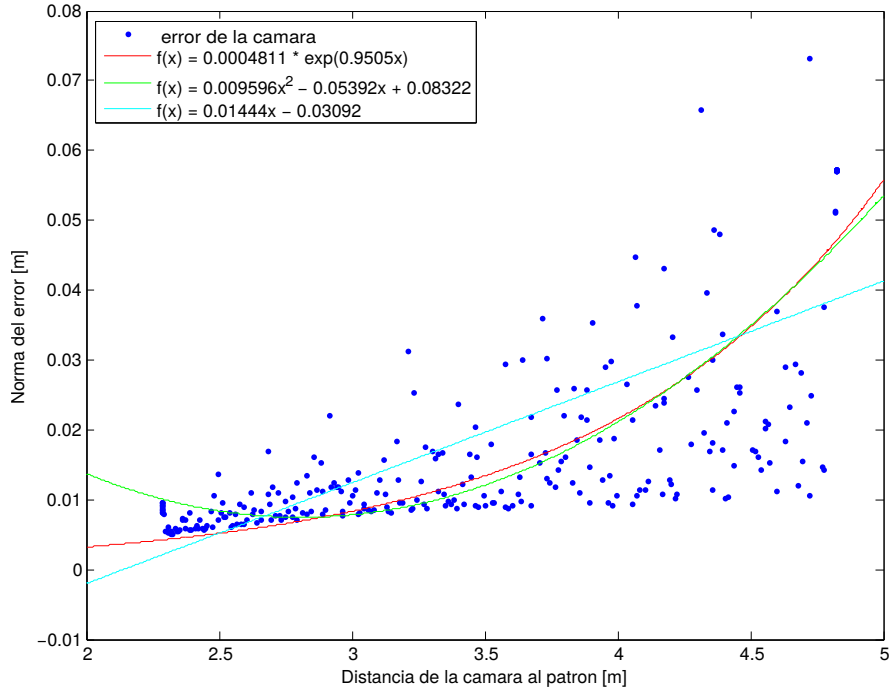


Figura 4.9: Error de localización en función de la distancia y funciones ajustadas sobre estos datos.

Se puede observar en primer lugar que la ecuación lineal no representa del todo a la distribución del error. En particular, para distancias menores a 2,1 m aproximadamente, el error esperado se hace negativo. Por otro lado, si bien la función cuadrática se asemeja a la exponencial a medida que crece la distancia, entre 2 m y 2,5 m ésta es decreciente y no es el comportamiento deseado. Finalmente, la función exponencial parece ser la que mejor ajusta a la distribución en general, por lo que se decidió utilizar esta última como modelo de error en los posteriores experimentos.

### 4.3. Ajuste del parámetro de filtrado de la velocidad

En la sección 3.2.2.2 se describe la forma en la cual se estima la velocidad utilizando un filtro pasa-bajos:

$$\mathbf{v}_t = \alpha (\mathbf{x}_t^{obs} - \mathbf{x}_{t-1}^{obs}) / \Delta t^{obs} + (1 - \alpha) \mathbf{v}_{t-1}$$

El objetivo de este experimento es encontrar el valor de  $\alpha$  tal que se minimice el ruido que se observa en la velocidad estimada. Para ello se realizó un experimento en el cual se utilizaron

distintos valores de  $\alpha$  y se evaluó el comportamiento del estimado de velocidad durante la ejecución del método.

La escena en este experimento es la misma que se utilizó para obtener el modelo de error en función de la distancia (figura 4.7). El procedimiento también es el mismo, es decir se trasladó al robot de una punta de la habitación hacia la otra a una velocidad constante de  $0,25\text{ m/s}$ . Los resultados pueden verse en la figura 4.10.

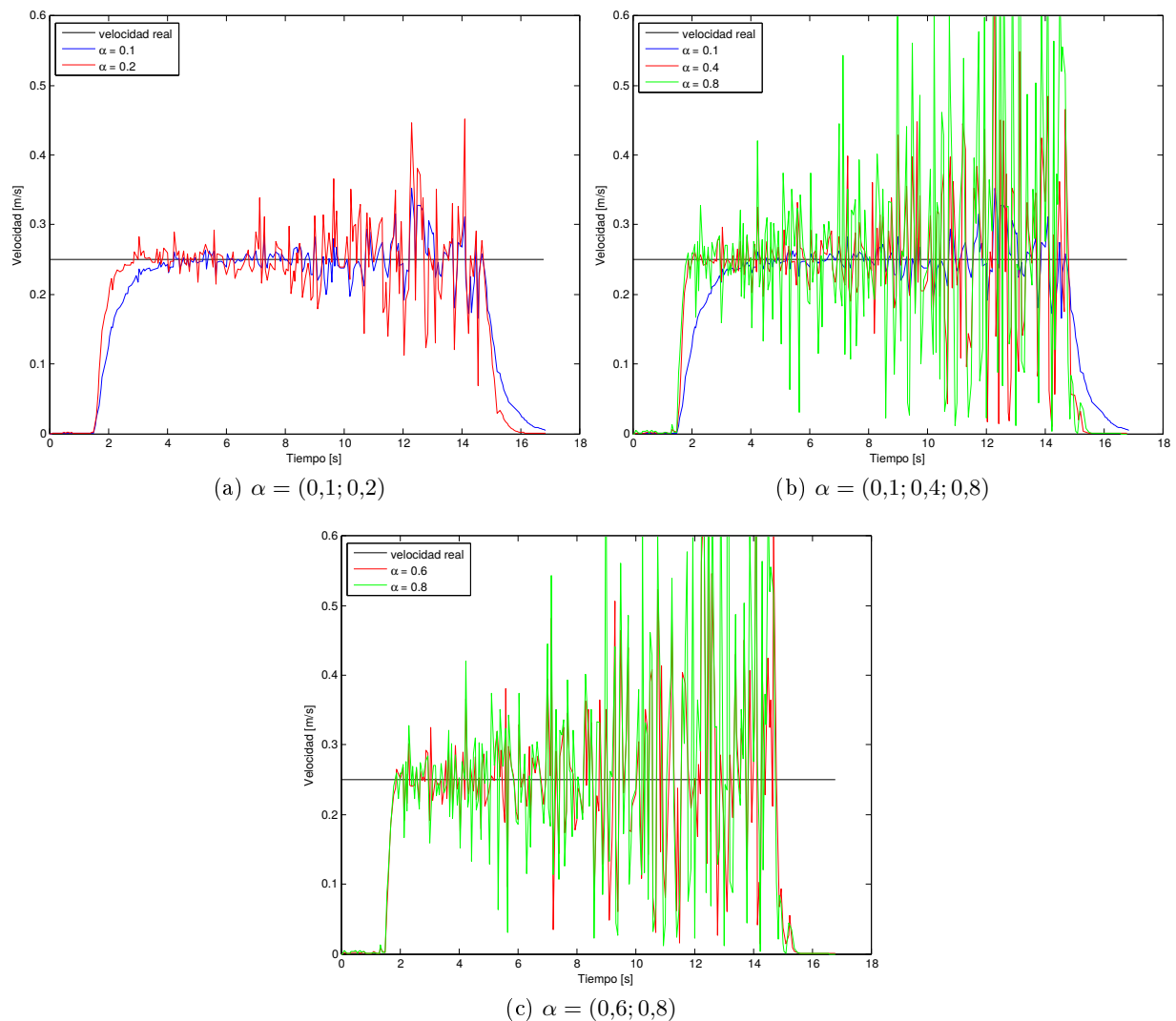


Figura 4.10: Velocidad en función del tiempo para determinar el mejor valor de  $\alpha$ . La velocidad real está marcada con negro en cada figura.

Para elegir el valor de  $\alpha$  se tienen en cuenta dos aspectos: el tiempo de convergencia a la velocidad real y la suavidad del estimado en comparación con la entrada del filtro. En este caso y en el resto de los experimentos el robot no realiza movimientos bruscos, por lo que no varía demasiado la velocidad en general. De esta forma, dentro del marco del experimento realizado, se deduce que  $\alpha = 0,2$  es un valor aceptable. Si bien con  $\alpha = 0,1$  el ruido es menor, se ve en los experimentos que tarda

considerablemente más en alcanzar la velocidad real. Por otro lado con  $\alpha$  más grande el ruido es mucho mayor y no hay una diferencia considerable en el tiempo de convergencia.

A partir de los resultados obtenidos, se puede observar que a medida que pasa el tiempo crece la diferencia entre la velocidad real y la estimada. Es algo esperable a partir del modelo, ya que al aumentar el tiempo del experimento, el patrón se aleja de la cámara, situación en la cual se sabe que el error de estimación de la posición del patrón aumenta.

#### 4.4. Análisis de precisión: máximo solapamiento con dos cámaras

El objetivo de este experimento es evaluar la precisión del método de localización en comparación con la precisión del método WhyCon al usar una sola cámara. Además, ambas estimaciones se contrastan con el *ground truth* del simulador. La hipótesis planteada es que al utilizar dos cámaras cubriendo toda la escena por la que el robot realiza el recorrido, es decir, con máximo solapamiento entre esas cámaras, la precisión aumenta con respecto a la utilización de una única cámara en la escena. Además, se quiere evaluar en este caso el sistema multicámara más chico, es decir con la menor cantidad de cámaras posibles, por ese motivo se utilizan únicamente dos en el primer experimento.

Para este experimento se diseñó una escena similar a la que se observa en los experimentos anteriores pero con dos cámaras enfrentadas, cada una en un extremo de la habitación con la inclinación necesaria para visualizar al patrón en todo momento. La figura 4.11 muestra la escena mencionada.

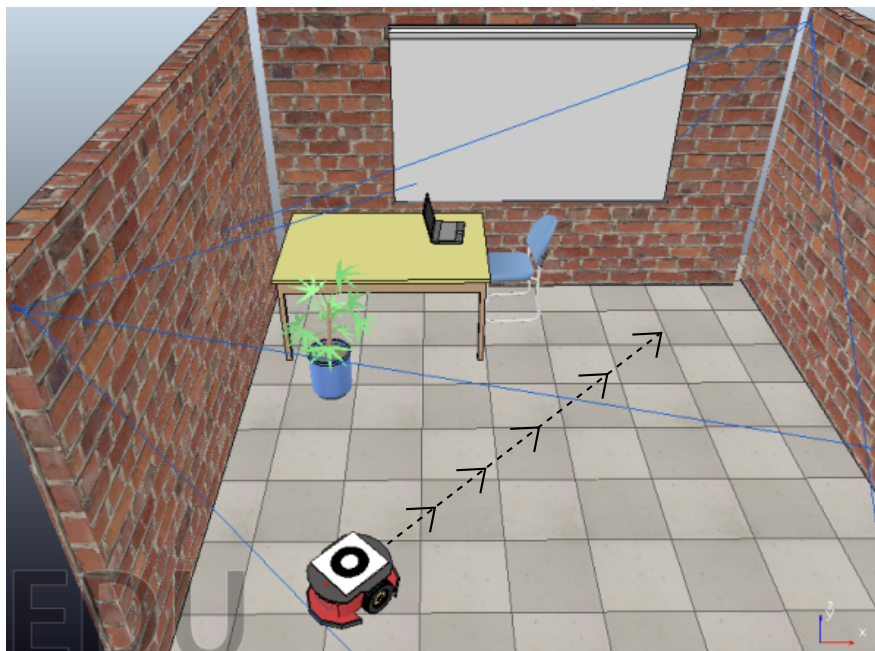


Figura 4.11: Escena con dos cámaras.

En la figura 4.12 se muestra el recorrido y la cobertura de cada cámara para entender mejor la

escena planteada.

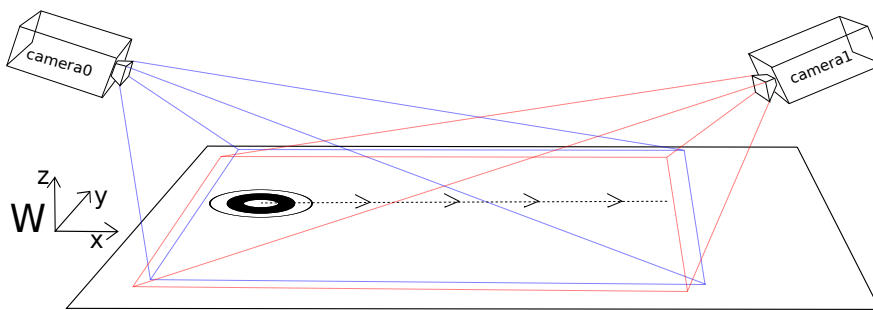


Figura 4.12: Esquema del recorrido en la escena en la cual el patrón se aleja de camera0 y se acerca a camera1.

En la figura 4.13 se presenta la norma del error de precisión respecto del *ground truth* para cada estimación. Asimismo en la figura 4.14 se presenta el error por cada eje. Por otra parte, para visualizar la distribución del error, se generaron histogramas de frecuencia de la norma del error que se muestran en la figura 4.15. Por último en la figura 4.16 se presenta la trayectoria estimada por el sistema de localización así como la información de *ground truth* y la estimación obtenida por cada instancia de WhyCon. Además cada 20 muestras se grafica la covarianza en  $XY$ , escalada por un factor que permite una mejor visualización.

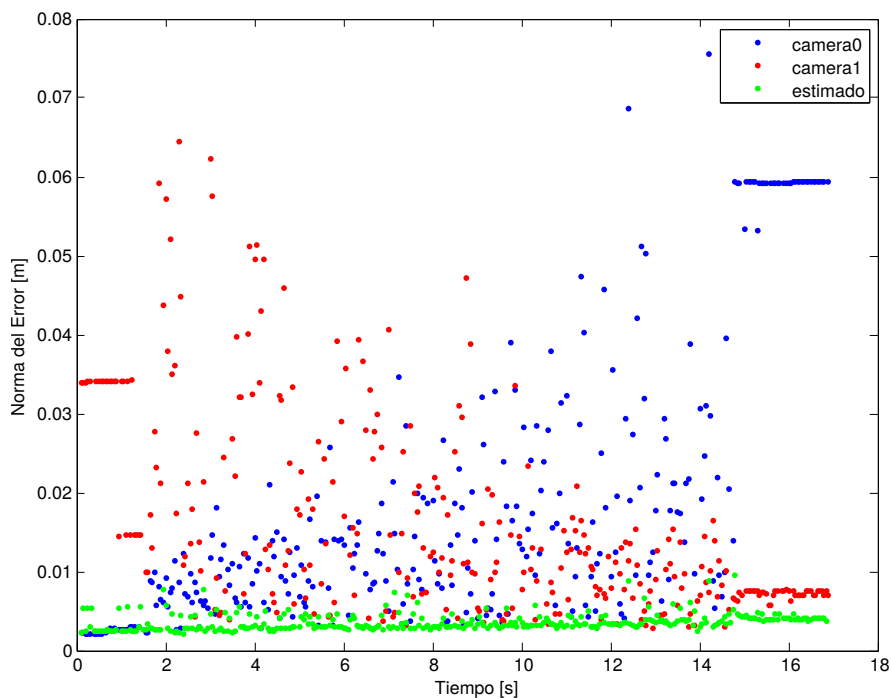
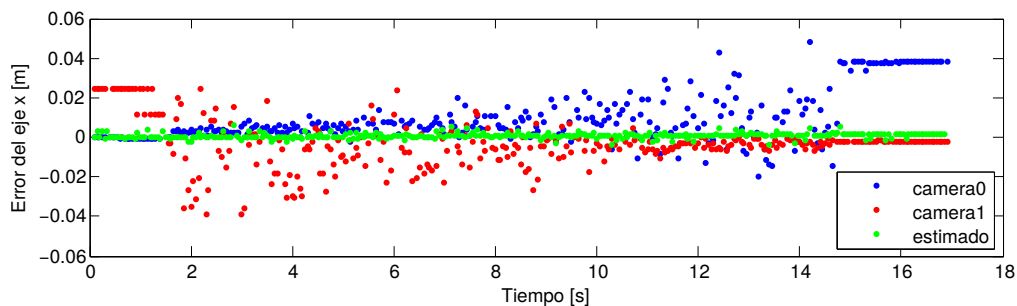
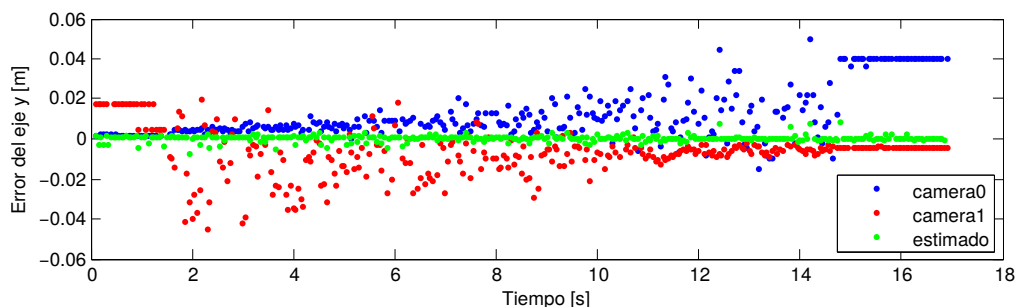


Figura 4.13: Error de estimación para el caso de dos cámaras con máximo solapamiento.

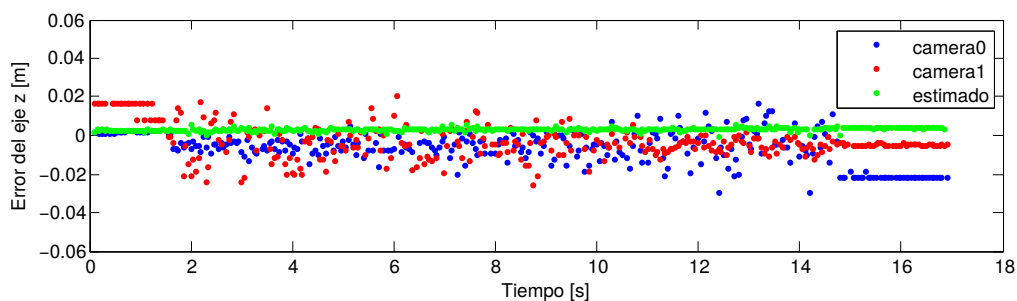




(a) Precisión en el eje X.



(b) Precisión en el eje Y.



(c) Precisión en el eje Z.

Figura 4.14: Precisión del método con dos cámaras en cada eje de coordenadas del mundo.

Analizando los resultados obtenidos en 4.13 y en 4.14, se puede observar como a medida que aumenta el tiempo el error crece para camera0 y decrece para camera1. Este comportamiento se atribuye a la distancia en cada momento del patrón a cada cámara, ya que a medida que avanza el tiempo, el patrón se aleja de camera0 y se acerca a camera1. Además, se puede ver que en todo momento la norma del error estimado es menor que cada instancia de WhyCon.

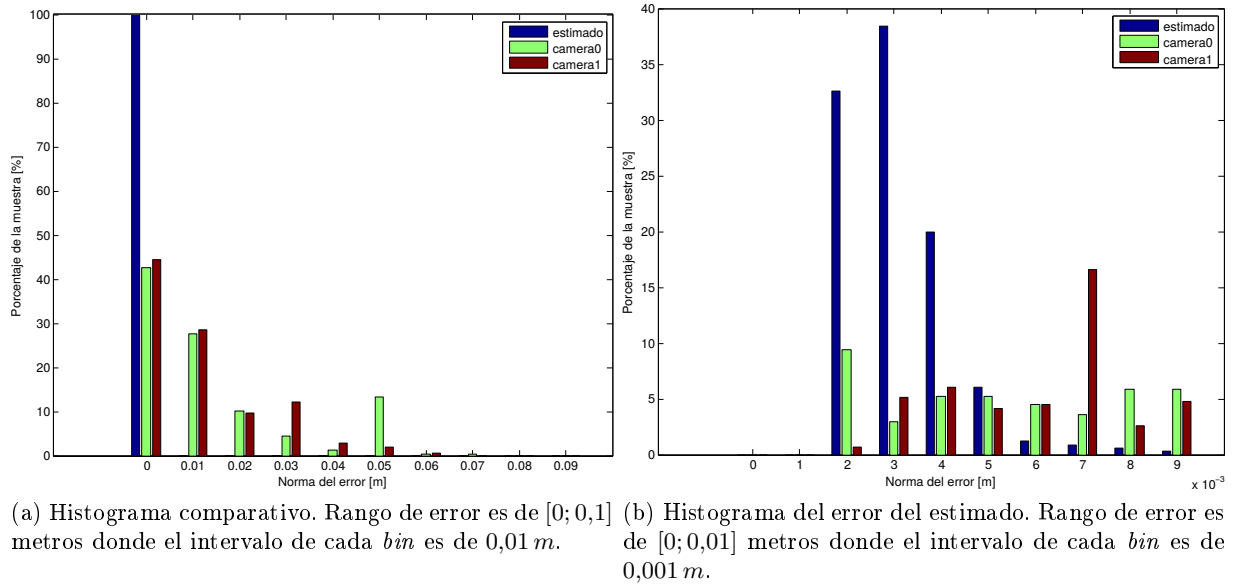


Figura 4.15: Histograma del error. Comparación con cámaras y ampliación del error de la posición estimada.

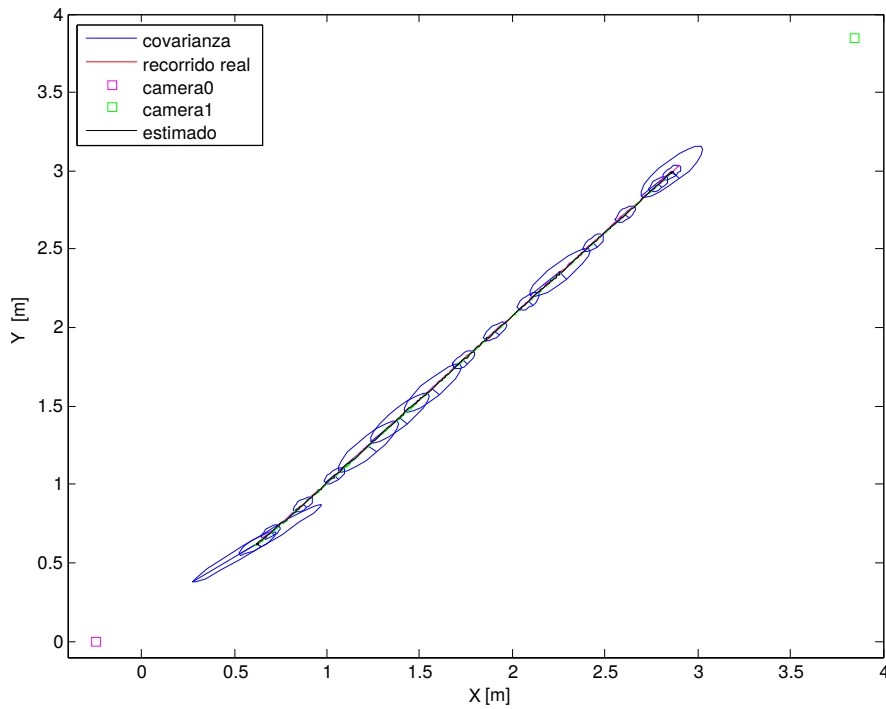


Figura 4.16: Trayectoria del método propuesto en contraste con el recorrido real y el de cada cámara. Las elipses azules muestran la covarianza por un factor de escala de 5000.

Por otra parte de los resultados obtenidos en 4.15a muestran como los valores de la norma del error del método entran todos en el primer *bin*, es decir menores a  $0,01$  m, a diferencia de las instancias de WhyCon de *camera0* y *camera1* que por más que tengan el pico de la distribución en el

*bin* de 0.01, poseen varias muestras de error más grandes. En la figura 4.15b se puede observar como la distribución de la norma del error del método tiene su pico en  $0,003\text{ m}$ , casi el 40% de la muestra tiene una norma de error de este valor. Por último, se puede destacar de los resultados obtenidos en 4.16 que la orientación de la covarianza se corresponde con la distancia del patrón a las cámaras. Las elipses que se ven más grandes corresponden a estimados producto de correcciones realizadas a partir de observaciones provenientes de la cámara que en ese instante de la trayectoria está más lejana mientras que las más pequeñas, corresponden a observaciones de la cámara más cercana.

#### 4.5. Análisis de precisión: máximo solapamiento con cuatro cámaras

El objetivo de este experimento es evaluar como varía la precisión del método al agregar cámaras en la escena. La hipótesis que se propone para este caso es que la precisión aumente al agregar más cámaras. Para poder analizar este caso se agregan dos cámaras adicionales a lo planteado en el experimento anterior, como se muestra en la figura 4.17.

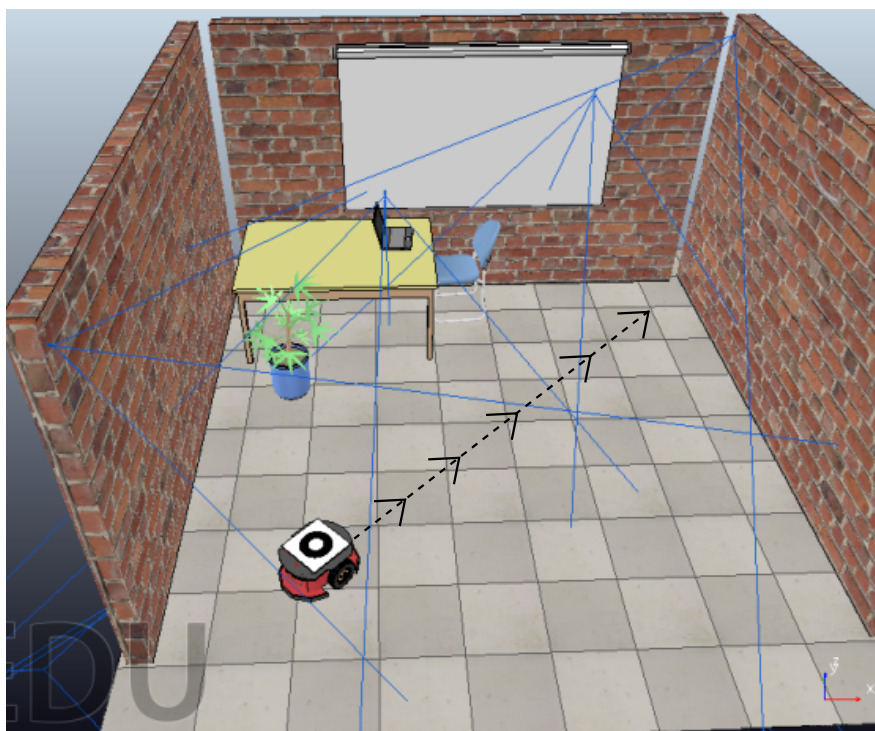


Figura 4.17: Escena con cuatro cámaras.

Las cámaras en las esquinas son iguales a las del caso previo, las cuales ven todo el recorrido en todo momento. Las cámaras que se encuentran en el techo, una de ellas ve la primera mitad del recorrido y la otra la segunda mitad. Entre estas dos últimas cámaras existe un mínimo solapamiento en la trayectoria del robot. De esta forma, en cada instante hay tres cámaras observando el patrón.

En la figura 4.18 se puede ver un esquema de este escenario.

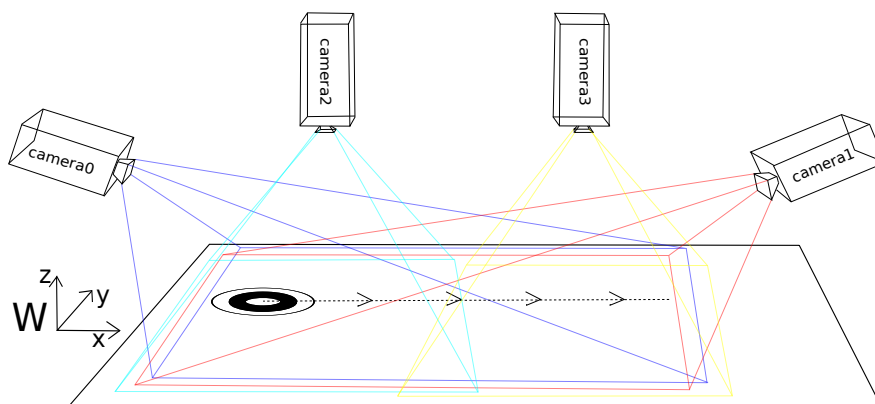


Figura 4.18: Esquema del recorrido en la escena en la cual el patrón se aleja de camera0 y se acerca a camera1, atravesando a su paso el trayecto cubierto por camera2 y camera3.

Se presentan a continuación los resultados obtenidos para este experimento. En la figura 4.19 se presenta la norma del error de precisión respecto del *ground truth* para cada estimación. En la figura 4.20 se presenta el error por cada eje. La distribución del error se puede observar en el histograma de frecuencia de la norma del error que se muestra en la figura 4.21. Por último, al igual que en el experimento con dos cámaras, se observa en la figura 4.22 la trayectoria del robot estimada por el método de localización propuesto junto a la matriz de covarianza.

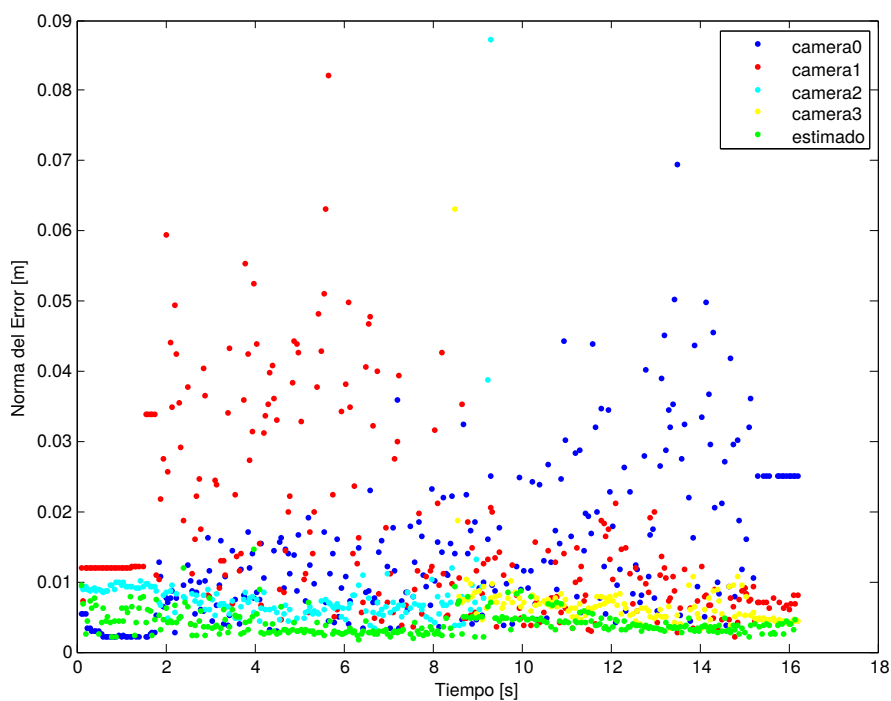
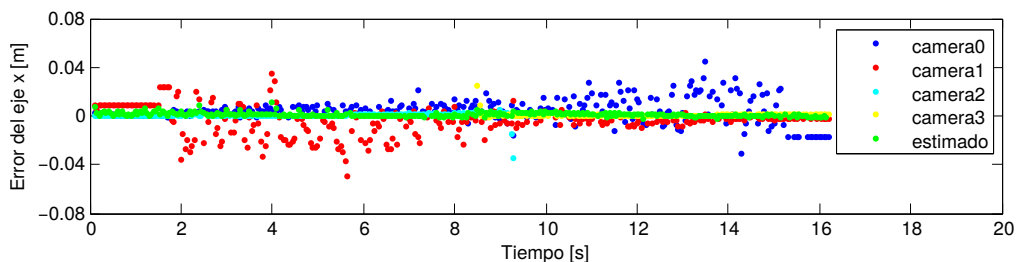
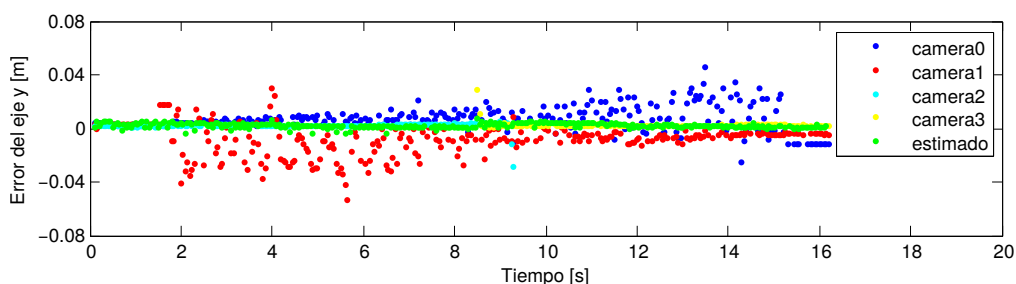


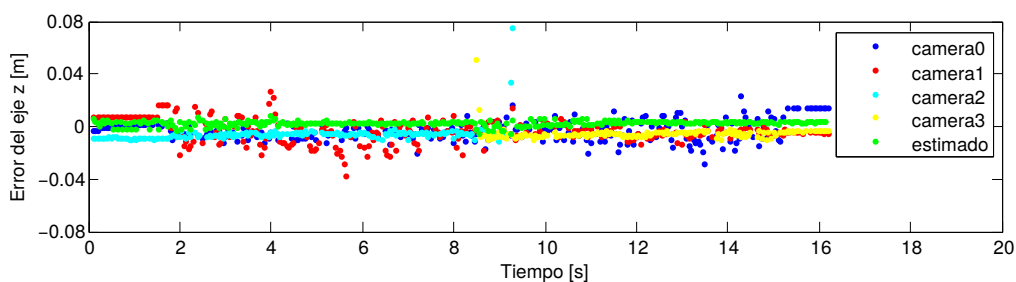
Figura 4.19: Error de estimación para el caso de cuatro cámaras.



(a) Precisión en el eje X.



(b) Precisión en el eje Y.



(c) Precisión en el eje Z.

Figura 4.20: Precisión del método con cuatro cámaras en cada eje de coordenadas del mundo.

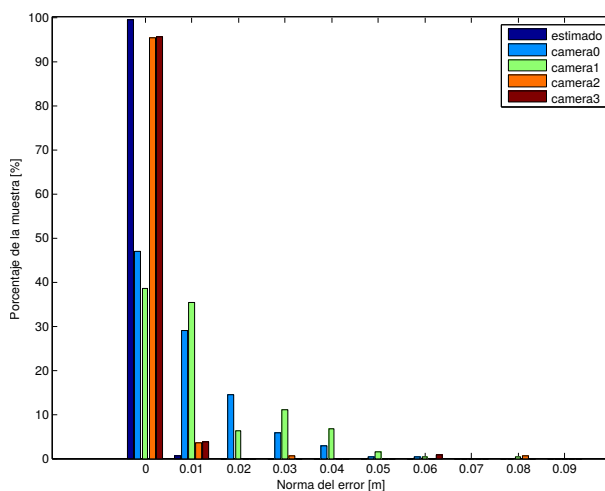


Figura 4.21: Histograma del error con cuatro cámaras. Rango de error es de  $[0; 0,1]$  metros donde el intervalo de cada *bin* es de  $0,01 m$ .

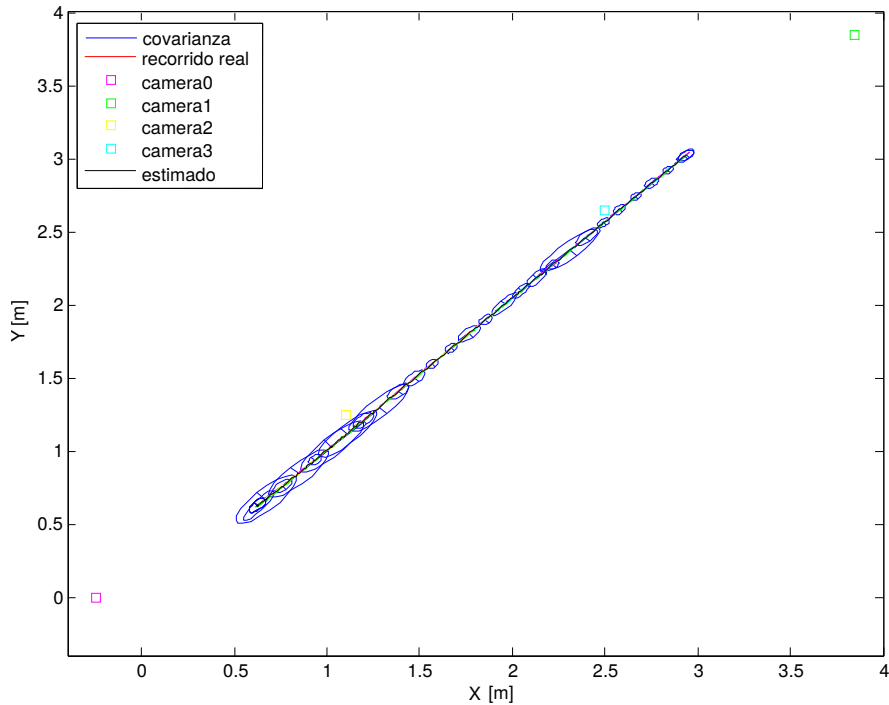


Figura 4.22: Trayectoria estimada por el método propuesto en contraste con el recorrido real y las cuatro cámaras. Las elipses azules muestran la covarianza por un factor de escala de 5000.

Evaluando los resultados de este experimento hasta el momento, se puede observar en la figura 4.19 y en la figura 4.20, como el comportamiento de la norma del error de las instancias de WhyCon asociadas a las cámaras en los extremos de la habitación es igual al del experimento anterior, como era de esperarse. Para las instancias de WhyCon asociadas a las cámaras del techo, la norma del error es menor que el de las otras dos y no tiene mucha variación, debido a que la distancia del patrón a estas cámaras es similar en todo el recorrido. También se puede observar en la figura 4.20 que el error de estas cámaras es mayormente en el eje  $Z$ , es decir en el eje vertical, que a su vez está alineado con el eje óptico de estas cámaras. Por último, en la figura 4.19, pueden observarse también ciertas muestras que poseen un error considerable (por ejemplo, camera2, segundo 9 aproximadamente). Estos casos corresponden a una situación particular en la cual el patrón queda parcialmente afuera de la imagen de la cámara y esto produce un error de localización por parte del sistema WhyCon. Esta situación se ilustra en la figura 4.23 donde se puede ver cómo difiere la estimación de la posición del patrón antes y después de llegar al borde. Debido a que no se visualiza en su totalidad el círculo negro genera una apariencia de tener un patrón más lejano, produciendo un error de localización. A su vez, estos errores influyen en la estimación del método de localización propuesto.



(a) Previo a llegar al borde.

(b) Sobre el borde.

Figura 4.23: Estimación de WhyCon con un patrón al borde de la imagen.

Por otra parte analizando los resultados de la figura 4.21, al igual que para el experimento anterior la mayoría de los errores de estimación del método caen por debajo de  $0,01 m$ .

Sobre los resultados presentados en la figura 4.22 se observan las distintas elipses que representan la covarianza del estimado en los ejes  $XY$ . Dependiendo de cual sea la observación de entrada al filtro  $EKF$  la elipse muestra una incerteza mayor cuando el patrón es observado por las cámaras de las esquinas, mientras que es menor cuando es observado por las cámaras del techo.

Por último, en busca de evaluar la precisión del método utilizando dos y cuatro cámaras, se comparan en la figura 4.24 los errores de estimación y en la figura 4.25 se muestra el histograma de frecuencia de error para cada caso.

Analizando estos últimos resultados comparativos se puede decir que al agregar más cámaras en la escena no necesariamente mejora la calidad del método. Como se observa en la figura 4.24, al estar la norma del error por debajo de  $0,01 m$  la comparación entre la cantidad de cámaras es similar y la variación se atribuye al ruido. Por otra parte, en la figura 4.25 se puede ver que el pico de la distribución para ambos casos se encuentra en  $0,003 m$ , es decir que la mayoría de los errores de estimación son entre  $0,003 m$  y  $0,004 m$ .

Estos resultados pueden atribuirse a que ya se contaba con una buena estimación con dos cámaras, que no se pudo mejorar agregando dos más. Sin embargo, puede verse que la incerteza es mucho menor cuando el patrón es observado por estas cámaras.

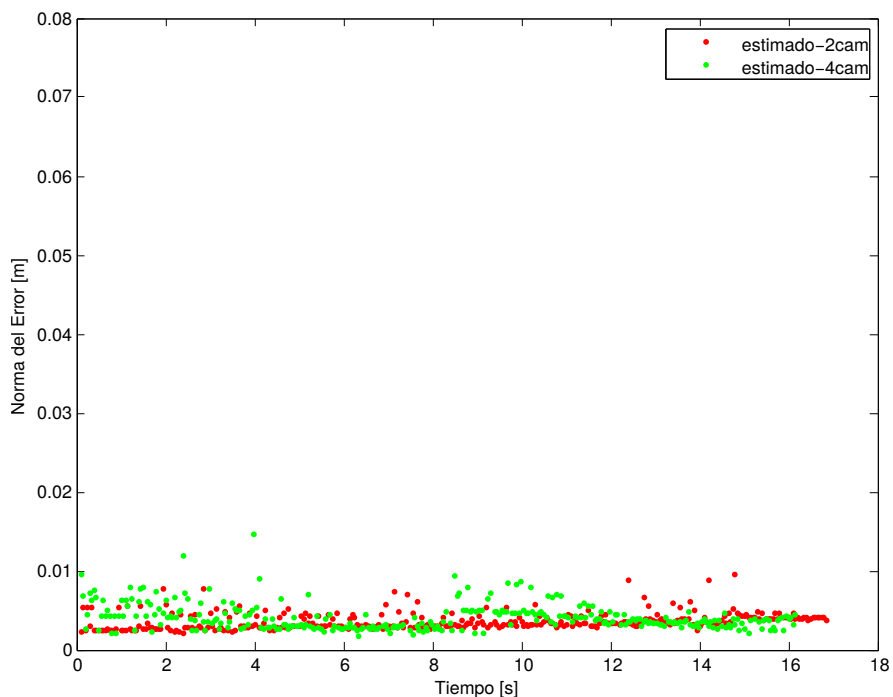


Figura 4.24: Error de estimación para el caso de cuatro cámaras y el caso con dos cámaras.

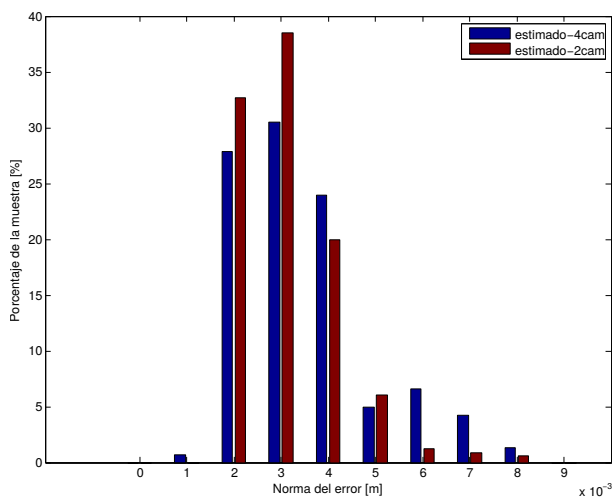


Figura 4.25: Histograma del error con cuatro cámaras y con dos cámaras. Rango de error es de  $[0; 0,01]$  metros donde el intervalo de cada *bin* es de  $0,001 m$ .



#### 4.6. Análisis de precisión: máxima cobertura con cuatro cámaras

A diferencia de los experimentos anteriores, en esta sección se desea analizar la precisión del método en una escena en la cual las cámaras están colocadas con el menor solapamiento posible con el objetivo de maximizar la cobertura de la trayectoria del robot.

Para realizar este experimento se diseña una escena con la forma de un pasillo largo en el que se colocan cuatro cámaras sobre el techo como muestra la figura 4.26. El patrón se encuentra sobre el robot de forma horizontal y el recorrido consiste en mover el robot de una punta a la otra del pasillo. El detalle del escenario se puede ver en la figura 4.27.

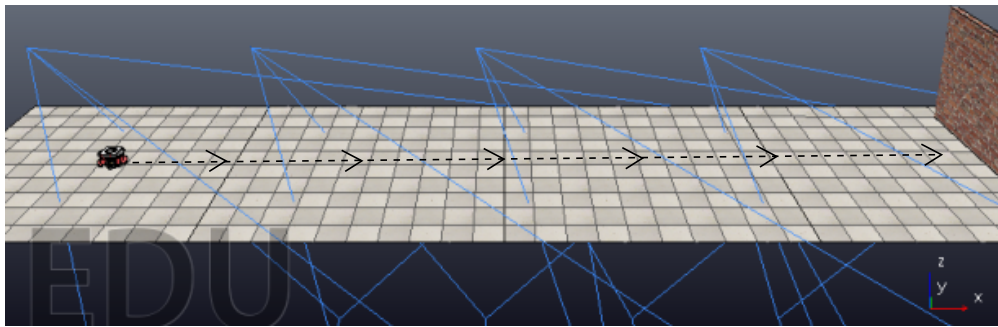


Figura 4.26: Escena pasillo con cuatro cámaras.

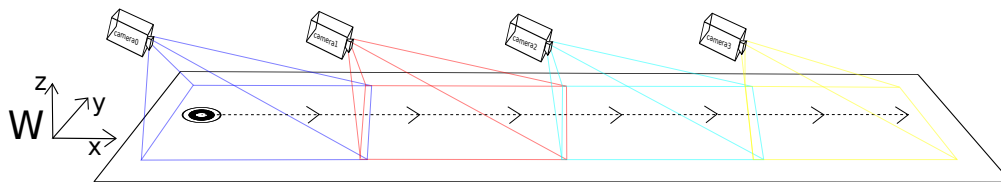
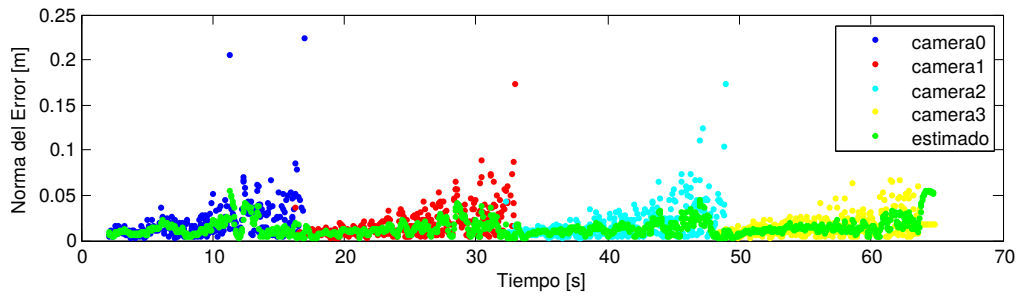
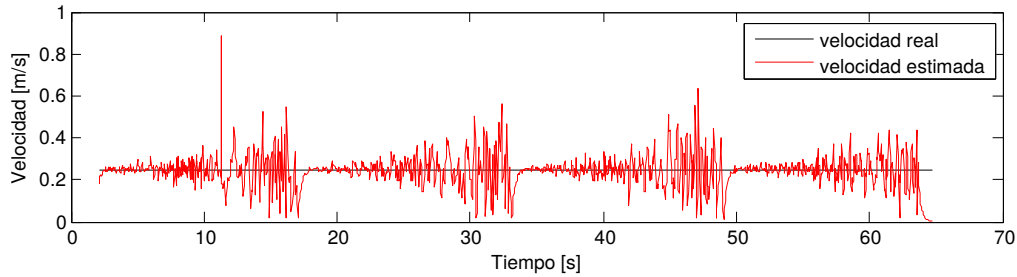


Figura 4.27: Esquema del recorrido en la escena en la cual el patrón llega al final del pasillo.

Para este experimento se realiza un análisis similar al de los experimentos anteriores. En la figura 4.28 se presenta la norma del error respecto del *ground truth* para cada estimación junto a la velocidad estimada por el método a lo largo del experimento. En la figura 4.29 se presenta el error por cada eje. La distribución del error se muestra en un histograma de frecuencia de la norma del error en la figura 4.30. Por último, en la figura 4.31 se puede observar la trayectoria del robot estimada por el método de localización propuesto junto a la covarianza del estimado en los ejes  $XZ$ , graficada cada 25 muestras.



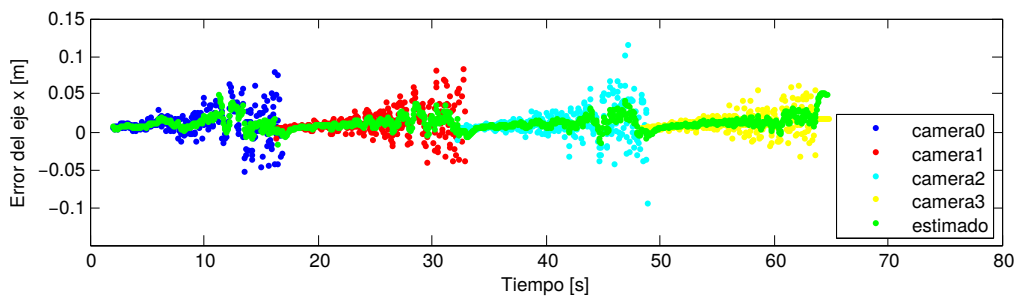
(a) Error de estimación.



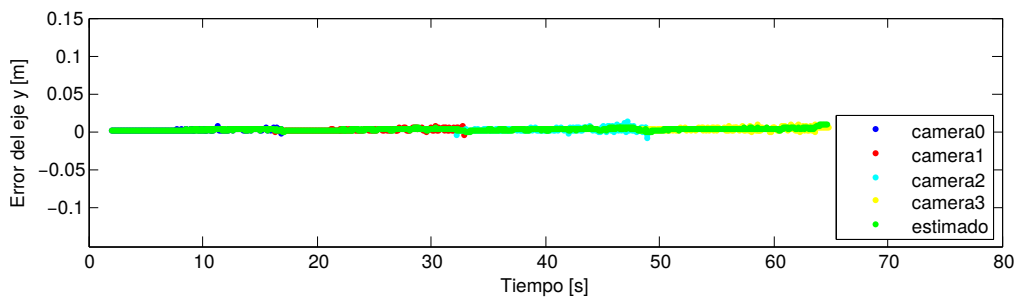
(b) Estimación de de la velocidad.

Figura 4.28: Velocidad estimada y errores de estimación para el caso con cuatro cámaras sobre el pasillo.

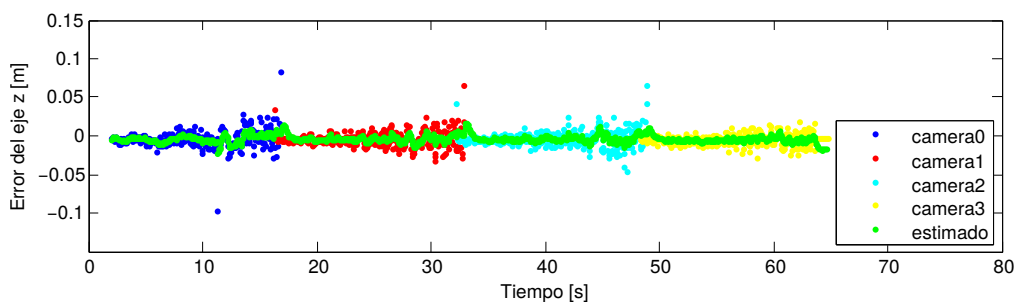
Se desprende de la figura 4.28a que el error de cada instancia de WhyCon crece al aumentar la distancia del patrón a la cámara. Asimismo, puede verse en la figura 4.28b que este error también aparece en la estimación de la velocidad. Como resultado de esta mala estimación de la velocidad, el filtro *EKF* produce también una mala estimación de la posición como producto del paso de predicción del filtro. Por otro lado, se observa en la figura 4.29b que los errores de estimación en el eje *Y* (dirección perpendicular del movimiento) son casi imperceptibles, tanto para las instancias de WhyCon de las cámaras como del método propuesto, debido a que prácticamente el patrón no se desvía en el sentido de este eje a lo largo del recorrido. Además la norma del error estimado está por debajo del error de cada instancia de WhyCon en cada instante de tiempo. Con respecto a la distribución del error de la figura 4.30, la mayoría de los errores que produce la estimación se encuentran por debajo de  $0,02\text{ m}$ . Sabiendo que el objetivo central del experimento no es la búsqueda de mejorar la precisión, en este caso no se puede concluir si la distribución del error estimado es mejor o peor que la distribución del error de las instancias de WhyCon. Esto se debe a que el método propuesto produce estimaciones a lo largo de todo el recorrido, a diferencia de cada instancia de WhyCon, que cubren únicamente el trayecto que alcanzan a visualizar por cada cámara. Por último, en la figura 4.31 se observa que la orientación y el tamaño de la covarianza en los ejes *XZ* se corresponden con la dirección en la que se encuentra la cámara con respecto al patrón, como era de esperarse a raíz de lo observado en los experimentos anteriores. Además, se puede ver en la figura 4.31 que en las regiones donde sí existe solapamiento (marcadas con línea punteada), la incerteza disminuye considerablemente ya que se cuenta con observaciones desde dos puntos de vista distintos.



(a) Precisión en eje X.



(b) Precisión en eje Y.



(c) Precisión en eje Z.

Figura 4.29: Precisión del método utilizando cuatro cámaras sobre el pasillo. Se mide la norma del error en función del tiempo, en cada eje de coordenadas del mundo.

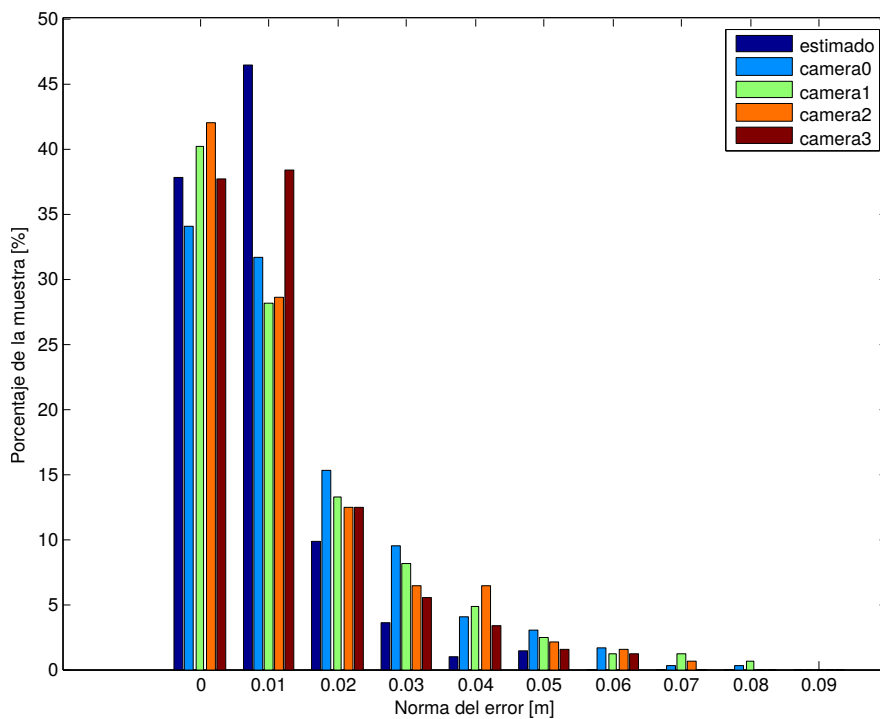


Figura 4.30: Histograma del error con cuatro cámaras sobre el pasillo. Rango de error es de  $[0; 0,1]$  metros donde el intervalo de cada *bin* es de  $0,01 m$ .

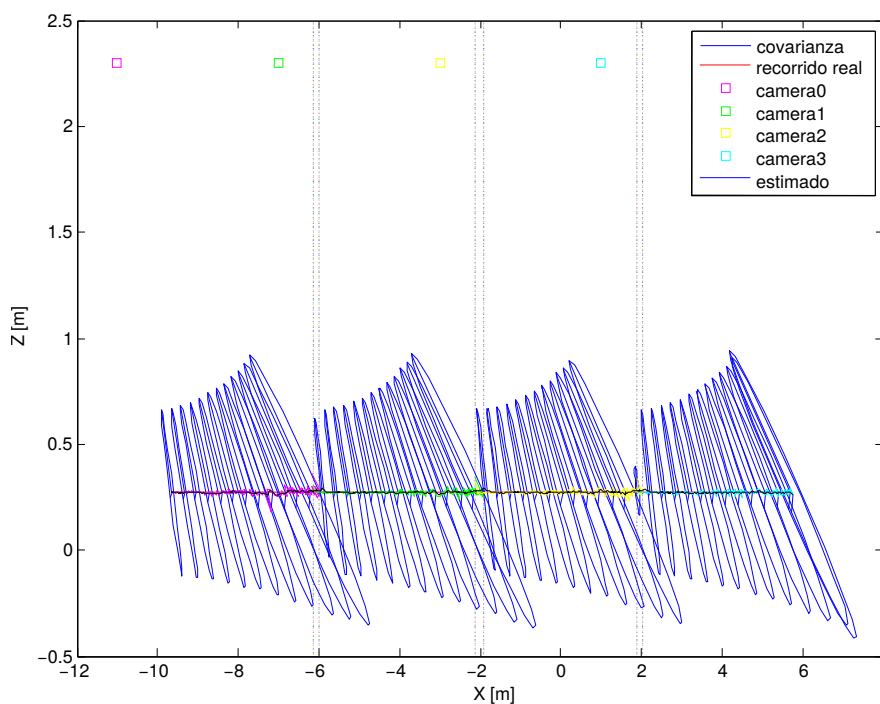


Figura 4.31: Trayectoria estimada por el método propuesto en contraste con el recorrido real y las cuatro cámaras por el pasillo. Las elipses azules muestran la covarianza por un factor de escala de 5000.

## 4.7. Análisis de precisión: escena general

Como último experimento, se propuso aplicar el conocimiento adquirido del desempeño del sistema en la práctica a un caso que se asemeje a una situación real, en contraste con una escena diseñada para evaluar una situación específica. Para ello se plantea un escenario el cual simula un piso de oficinas como se muestra en la figura 4.32. La distribución de las cámaras se diseñó en base a los resultados de los experimentos previos intentando cubrir todo el espacio posible sin sobrecargar de cámaras el lugar, en lo que se consideró la mejor ubicación para cada una de ellas. En las oficinas se colocó una cámara en cada esquina para cubrir todo el interior de la misma con gran precisión. Sobre el pasillo se colocaron cuatro cámaras a lo largo del mismo cubriéndolo completamente. En el salón más grande se colocaron cuatro cámaras cubriendo el lado izquierdo, otras cuatro cubriendo el lado derecho y una adicional que cubre la entrada. En total se distribuyeron 29 cámaras a lo largo de toda la escena.

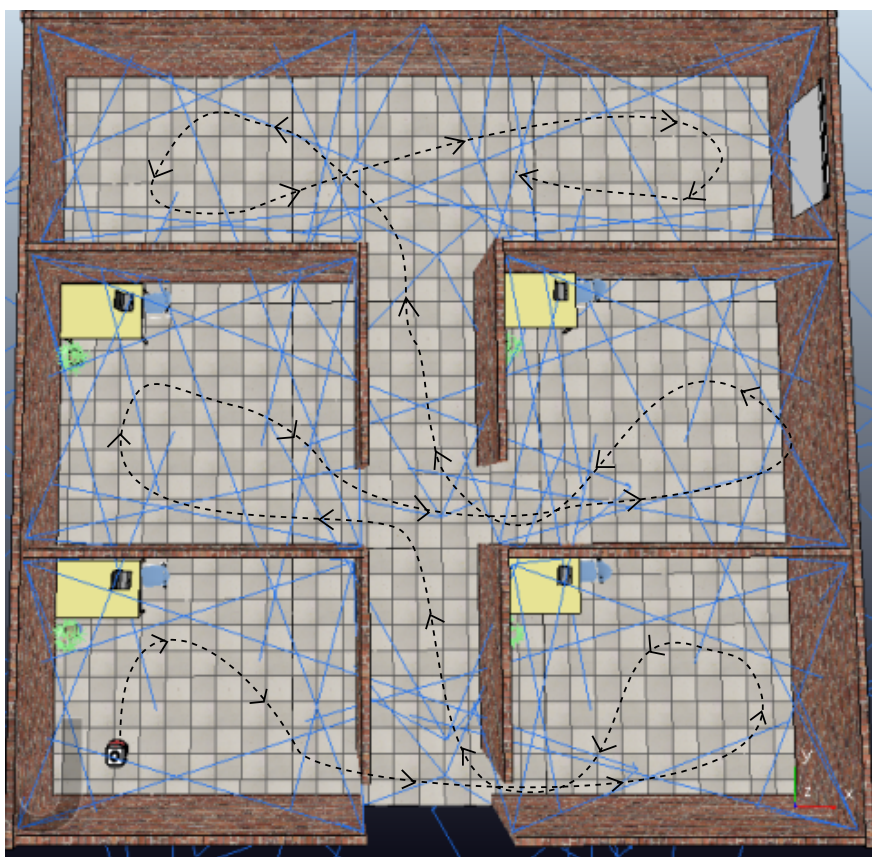


Figura 4.32: Escena que simula un piso de oficinas. El recorrido del robot está marcado con línea punteada.

La trayectoria realizada por el robot cubre toda la escena y todas las cámaras captan al robot en algún momento del recorrido, sin embargo hay ciertos instantes de tiempo en los cuales el patrón no alcanza a ser visto por ninguna cámara.

Como primer resultado, se presenta en la figura 4.33 la trayectoria estimada por el método propuesto junto a la trayectoria real informada por el simulador. A su vez, se incluyen las trayectorias individuales estimadas por cada instancia de WhyCon. Además, cada 25 muestras y aumentadas por un factor de escala, se graficó la covarianza del estimado.

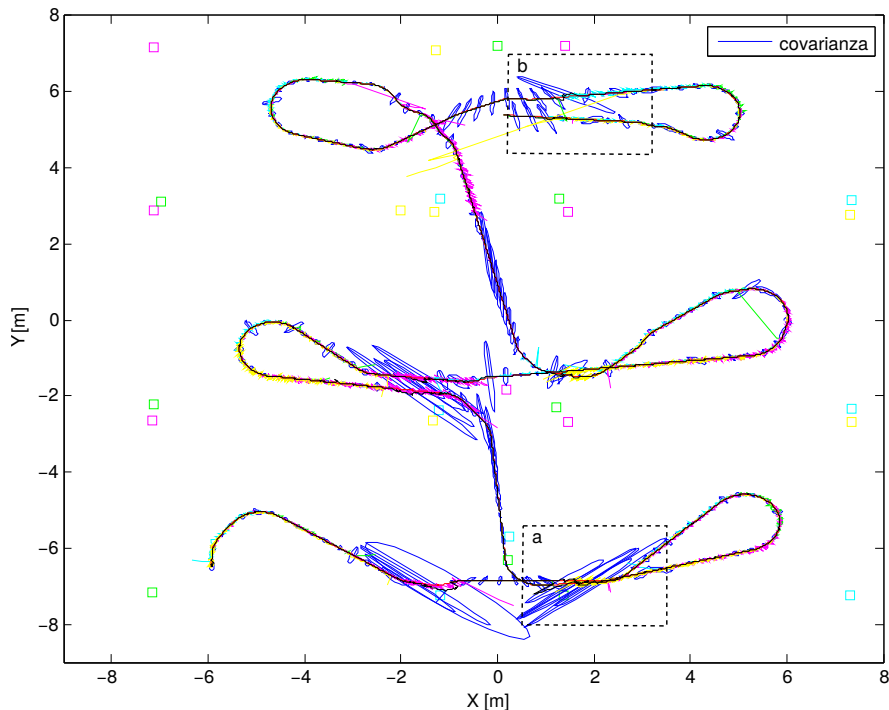


Figura 4.33: Recorrido del robot sobre el escenario de oficinas. Las elipses azules muestran la covarianza del estimado en los ejes  $XY$ , aumentada por un factor de escala de 5000. El recorrido real se encuentra en rojo. Por encima de éste se ve el de cada cámara en diferentes colores. Por sobre estos se muestra en negro el recorrido estimado por el método.

En primer lugar, se puede ver en la figura 4.33, que la trayectoria estimada es muy similar a la real. Además, la incerteza por lo general es baja, exceptuando algunas áreas donde las cámaras están lejanas al patrón o donde el sistema deja de recibir observaciones por no haber cámaras que observen el patrón y entonces realiza predicciones de la pose, en estos casos la covarianza crece. Dos de estas áreas están marcadas en la figura y serán analizadas posteriormente.

Como en los experimentos anteriores, en este caso se desea observar la precisión del método propuesto comparando la posición estimada con la posición real del patrón. Por ello, en la figura 4.34 se muestra el valor de la norma del error a lo largo del tiempo.

En concordancia con los resultados de la trayectoria estimada que se presentaron anteriormente, en la figura 4.34 se observa un error que por lo general está debajo de los  $0,02\text{ m}$ . Por otro lado, se pueden observar algunos picos, los cuales corresponden a ciertas etapas de la trayectoria en las cuales ninguna cámara puede observar el patrón, o a casos en los cuales la cámara se encuentra a una distancia bastante lejana como se mencionó anteriormente.

Para ampliar el análisis sobre la distribución del error de la muestra, se generaron histogramas de frecuencia de la norma del error. Esto se puede ver en la figura 4.35. Se observa en los histogramas que la mayoría de los errores del método caen por debajo de  $0,01\text{ m}$ . Para ser más precisos, la distribución del error posee un pico en  $0,004\text{ m}$ .

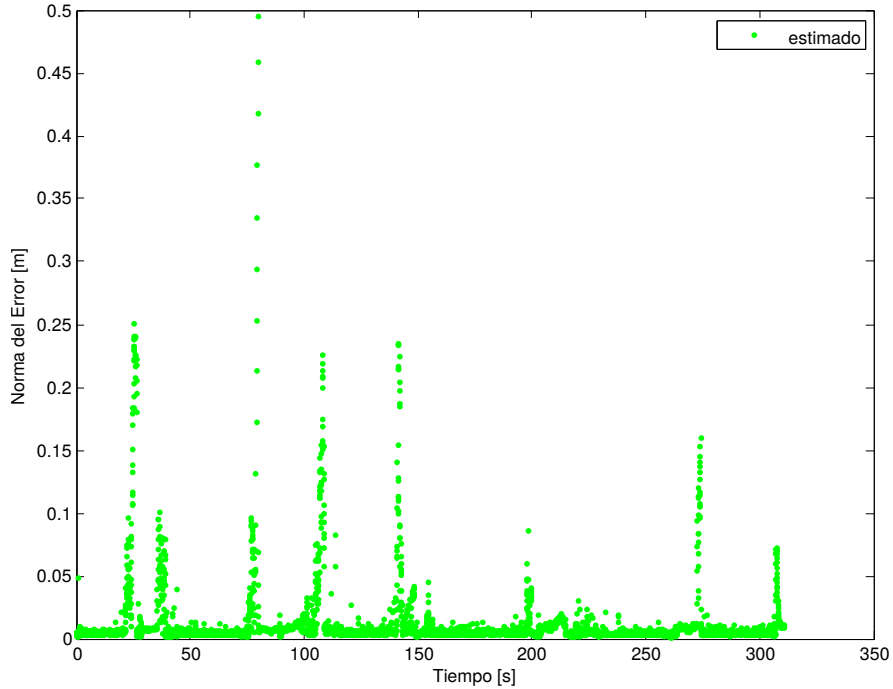
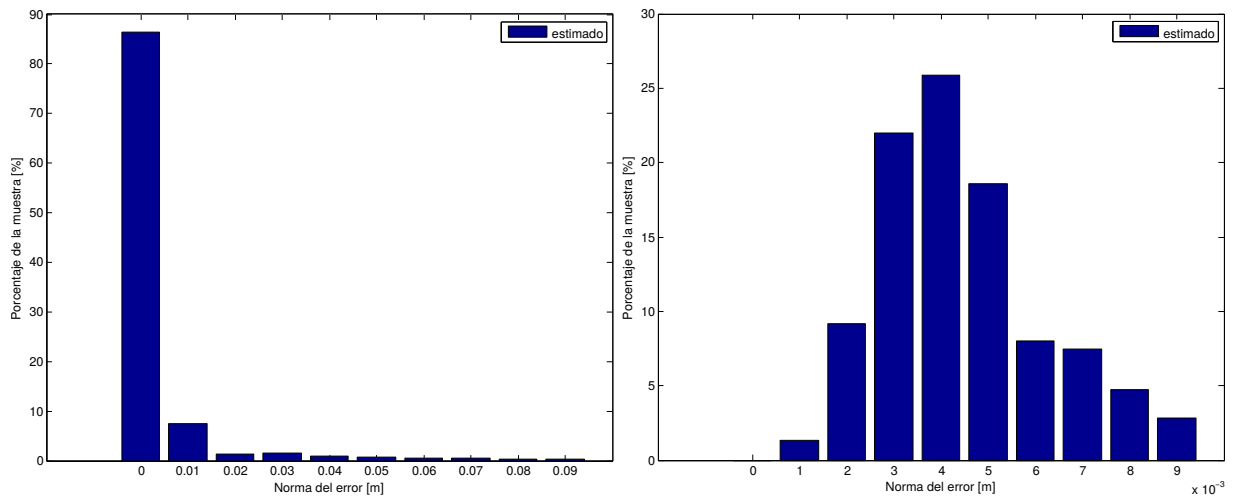


Figura 4.34: Error de estimación para el caso de la escena general.



(a) Histograma del error del estimado. Rango de error es de  $[0;0,1]$  metros donde el intervalo de cada *bin* es de  $0,01\text{ m}$ .  
 (b) Histograma del error del estimado. Rango de error es de  $[0;0,01]$  metros donde el intervalo de cada *bin* es de  $0,001\text{ m}$ .

Figura 4.35: Histograma del error de la escena grande.

Por último, sobre este experimento se obtuvieron adicionalmente algunos resultados generales para tener en cuenta. El promedio de la norma del error estimado es de  $0,012\text{ m}$  y el máximo valor que tiene la norma del error es de  $0,496\text{ m}$ . Por otra parte, la máxima distancia del patrón a una cámara en una observación es de  $10,016\text{ m}$  y en dicha observación la norma del error de la instancia de WhyCon sobre esa cámara es de  $1,174\text{ m}$ , cuando la norma del error estimado en ese mismo instante es de  $0,011\text{ m}$ .

## 4.8. Análisis de casos particulares

En esta sección, basándose en la misma escena del experimento anterior se buscó analizar casos particulares a lo largo del recorrido. El objetivo de estos casos es analizar la situación en la que el filtro *EKF* no recibe observaciones de las instancias de WhyCon y solo puede realizar predicciones de la pose.

La primer ampliación, figura 4.36, es sobre el sector (a) de la figura 4.33, en la cual se puede observar que la covarianza crece considerablemente en un momento determinado del recorrido.

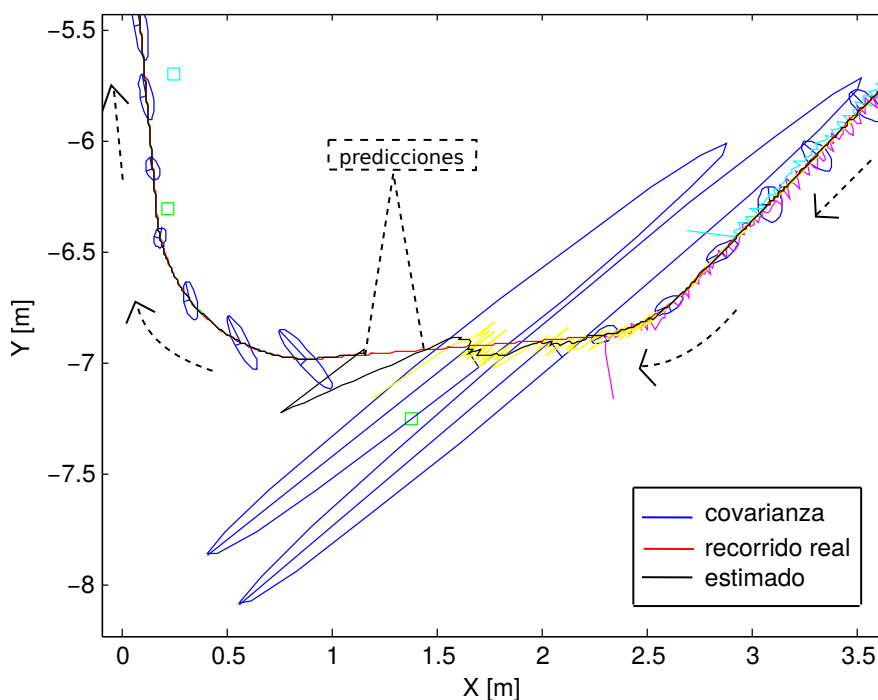
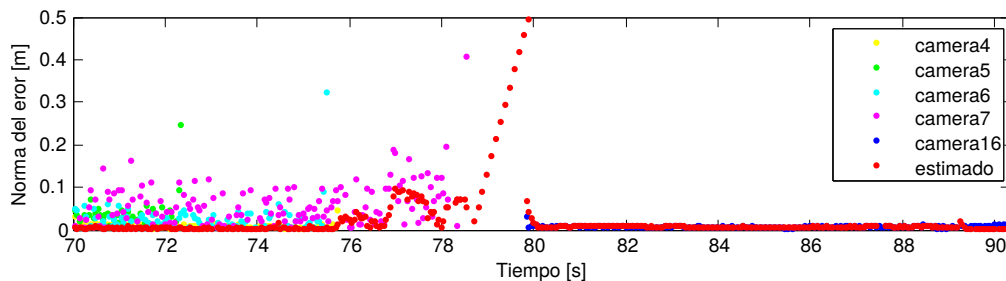


Figura 4.36: Recorrido del robot sobre escena grande ampliación (a). Por encima del recorrido real se ve el de las cámaras en diferentes colores.

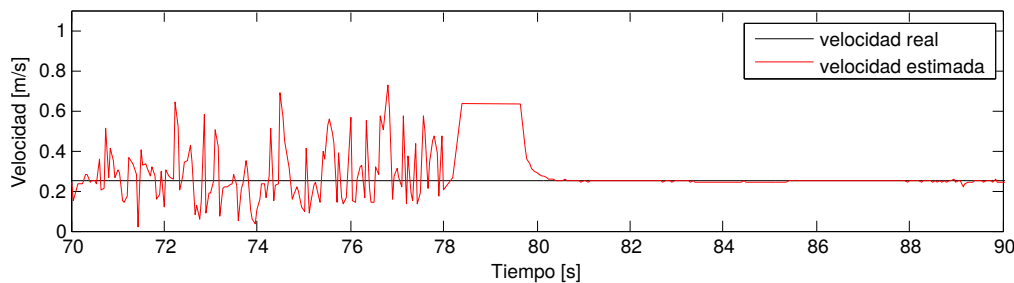
En esta primer ampliación se puede observar un intervalo de tiempo (marcado en la figura 4.36) durante el cual el filtro no recibe observaciones y realiza únicamente predicciones. En este caso se puede ver un error bastante pronunciado. Esto se debe a que las últimas observaciones previas a



este intervalo son de una instancia de WhyCon asociada a una cámara lejana. Notar que justo antes del tramo mencionado la elipse de incertidumbre es bastante grande y alargada en dirección a esta cámara en cuestión. Este error en la estimación de WhyCon provoca que la velocidad estimada también tenga error y por ende, el error de las predicciones de la pose es grande. Por otro lado, se puede ver que luego de esto, al recibir una nueva observación se corrige el recorrido inmediatamente como producto del paso de corrección del filtro. Para observar en detalle el comportamiento del método y de las cámaras sobre este tramo se muestra en la figura 4.37 el error de estimación del método y de las cámaras involucradas y la velocidad estimada por el método para comparar con los errores.



(a) Error de estimación del método.



(b) Velocidad estimada.

Figura 4.37: Error de estimación del método sobre la ampliación (a) y la velocidad estimada.

En la figura 4.37a se puede observar como hasta los 76 s la norma del error estimado es chica debido a que hay varias cámaras viendo al patrón. Luego, entre 76 s y 78 s crece el error, en este caso porque hay una única cámara observando la escena y está lejos del patrón. Entre 78 s y 80 s se observa el instante de tiempo en el cual el método realiza únicamente predicciones con un error creciente. Notar que en el mismo intervalo de tiempo, en la figura 4.37b la velocidad permanece constante y con un error grande, proveniente de la mala estimación anterior, lo que provoca que el error en la predicción también sea grande. A partir de los 80 s, el método recibe observaciones de una cámara cercana que casi no posee errores de estimación y el método de localización corrige la posición estimada y la velocidad.

El segundo caso a analizar corresponde al marcado como (b) en la figura 4.33 y se presenta en la figura 4.38, en la cual se puede observar otro tramo donde el método debe estimar la trayectoria únicamente mediante predicciones.

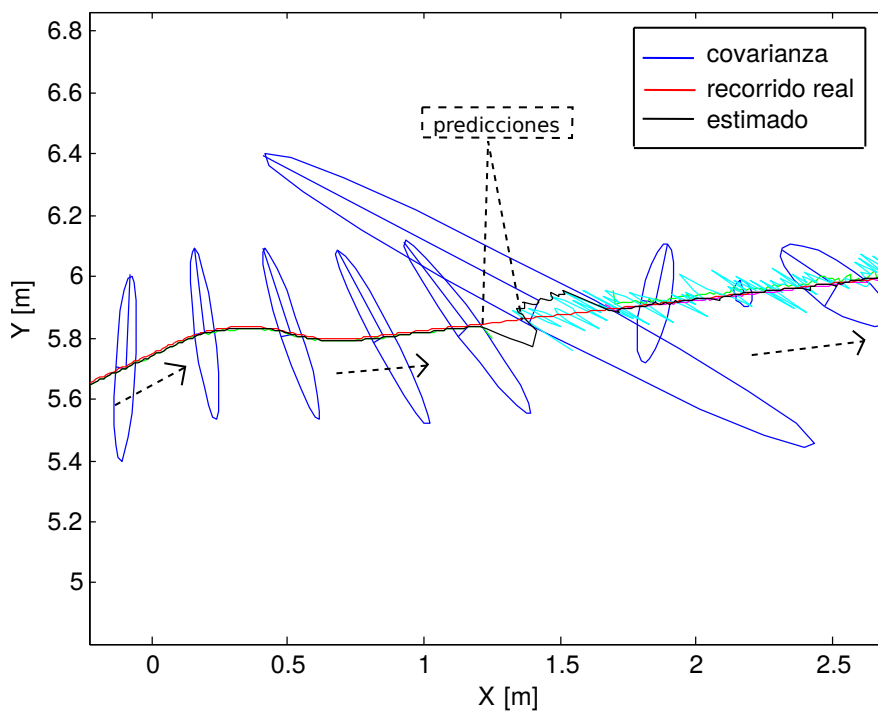
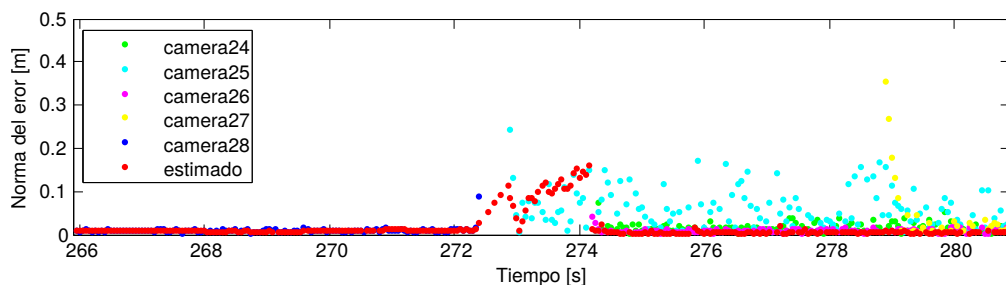
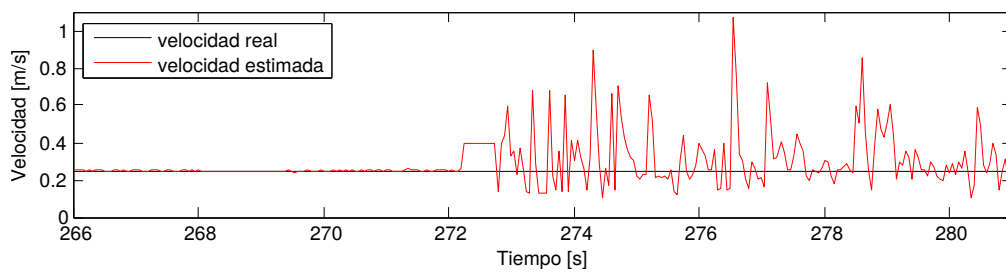


Figura 4.38: Recorrido del robot sobre escena grande ampliación (**b**). Por encima del recorrido real se ve el de las cámaras en diferentes colores.

En este caso se puede observar cómo la estimación del método casi no se diferencia de la trayectoria real, dado que se cuenta con observaciones de una cámara que se encuentra cercana al patrón. Al dejar de recibir estas observaciones, el método continúa estimando la posición del patrón pero a través de predicciones. Por esta razón, el error es mucho menor en comparación con el caso anterior. Luego de este intervalo, se puede observar un error grande debido a que la única cámara que comienza a ver al patrón se encuentra bastante alejada. Esto explica el tamaño de la covarianza hasta recibir observaciones de más cámaras que mejoran la estimación del método propuesto. Por último, para completar el detalle de la ampliación (**b**) se presenta la figura 4.39a en la cual se observa el error de estimación en función del tiempo del método de localización y de las cámaras involucradas, y la figura 4.39b en la cual se observa la velocidad estimada por el método propuesto.



(a) Error de estimación del método.



(b) Velocidad estimada.

Figura 4.39: Error de estimación del método sobre la ampliación (b) y la velocidad estimada.

Nuevamente en este caso se puede observar un intervalo de tiempo (a partir del segundo 272) donde el método únicamente realiza predicciones, donde se observa que la velocidad estimada difiere bastante de la real.

## Capítulo 5

# Conclusiones

En este trabajo se presenta un método de localización externa basado en visión utilizando múltiples cámaras que permite obtener la pose de un robot autónomo móvil, mediante la estimación individual obtenida por cada cámara, utilizando el Filtro de Kalman Extendido. Para las estimaciones individuales se emplea WhyCon, un sistema muy utilizado para obtener información de *ground truth*. WhyCon utiliza cámaras de bajo costo como sensores y es capaz de procesar cientos de imágenes por segundo con una gran precisión y gran robustez. Sin embargo este sistema posee ciertas limitaciones, siendo que está basado en una única vista de la escena, el área de cobertura está limitada únicamente al alcance de la cámara. Por otro lado, se sabe que la precisión del método está fuertemente ligada a la distancia del patrón a la cámara. Estas limitaciones motivan el desarrollo propuesto en este trabajo.

En primer lugar, se propuso un modelo del sistema de localización multicámara donde el objetivo del mismo es fusionar la información proveniente de múltiples instancias de WhyCon y así mejorar la precisión de las estimaciones.

A continuación, se presentaron dos formas de calibración de este sistema, una utilizando la herramienta Kalibr y otra, utilizando el mismo WhyCon, cada una con sus ventajas y desventajas. Por su parte, Kalibr permite obtener tanto los parámetros intrínsecos como extrínsecos simultáneamente. Sin embargo es un sistema que no se puede utilizar en tiempo real. WhyCon, por otra parte, es el sistema que de por sí se utiliza como entrada del sistema de localización propuesto, por lo que resulta una herramienta conocida para el usuario. Además, el proceso de calibración es mucho más simple y funciona en tiempo real. Sin embargo, solamente proporciona los parámetros extrínsecos de cada cámara, por lo que se debe calibrar cada una por separado para obtener los parámetros intrínsecos.

Habiendo presentado el modelo del sistema propuesto, se presenta el método de fusión utilizado, que corresponde al Filtro de Kalman Extendido. De esta forma, se proponen los modelos utilizados por el filtro, es decir: el modelo de movimiento, el modelo de observaciones y los modelos de error. En cuanto a estos últimos se destacó el estudio del modelo de error en función de la distancia el cual proviene de las instancias de WhyCon y que permitió optimizar el uso de la información redundante en pos de mejorar la precisión.

Una vez definido completamente el método propuesto, se realizaron distintos tipos de experimentos utilizando el simulador V-Rep, en los cuales se compararon los resultados de localización obtenidos con la información de *ground truth*.

Previo a la evaluación del sistema propuesto, se evaluaron las dos formas de calibración previamente propuestas. En este sentido, se pudo comprobar la factibilidad de uso de ambos sistemas, lo que sería un paso necesario al afrontar experimentos con cámaras reales.

Como segundo paso previo a los experimentos sobre el sistema final, se buscó encontrar experimentalmente un modelo de error para el sistema de localización WhyCon. Se dedujo de los resultados obtenidos mediante el ajuste de las muestras del error de una instancia de WhyCon en función de la distancia al patrón, que la función que mejor se adecúa es una función exponencial. Por otro lado, se realizó también un experimento con el fin de obtener el valor del parámetro de filtrado del cálculo de la velocidad tal que minimice el ruido de estimación de dicha variable.

Habiendo encontrado experimentalmente los parámetros y modelos necesarios, se procedió a analizar el desempeño del sistema de localización en diversas configuraciones estereotípicas. En el primer experimento, se buscó maximizar el solapamiento de las cámaras bajo la hipótesis de que se lograría maximizar la precisión. Para ello se diseñaron escenas en las cuales se utilizaron cámaras convergentes, donde todas ellas ven la misma escena. Los resultados obtenidos en estos experimentos muestran que el promedio de error del método de localización es de  $0,003\text{ m}$ , y todos los errores están por debajo de  $0,01\text{ m}$ , cuando para las instancias de WhyCon hay errores de hasta  $0,08\text{ m}$ . Además, en cada instante de tiempo, por lo general el error estimado está por debajo de los errores de cada instancia de WhyCon asociada a las cámaras. Paralelamente, a raíz de estos experimentos se pudo detectar que el sistema WhyCon introduce un error al método propuesto cuando el patrón se encuentra en el borde de la imagen observada. Evitando este inconveniente probablemente mejoraría la estimación del método en estas situaciones.

Como segundo experimento, se buscó maximizar la cobertura del recorrido del robot, dado que esto corresponde a un caso de uso deseado del sistema propuesto. De esta forma, se planteó un escenario simulado donde cuatro cámaras con mínimo solapamiento permitieron cubrir una región de  $18\text{ m}$  de largo, sin perder de vista en ningún momento al patrón. De los resultados obtenidos se pudo observar que incluso en este caso, el error de localización fue menor a  $0,05\text{ m}$ . En particular, si bien el estimado es casi en su totalidad producto de las observaciones de una sola cámara, el uso del filtro *EKF* permitió minimizar el ruido de estimación a partir de utilizar el modelo de error propuesto.

Finalmente, aplicando el conocimiento adquirido sobre el desempeño del sistema durante la etapa de una primera experimentación, se volcó lo aprendido para diseñar una escena simulada de grandes dimensiones semejante a la realidad. A partir de los resultados obtenidos luego de realizar este experimento, en primer lugar se puede mencionar que el funcionamiento del sistema fue en general muy bueno y acorde a las expectativas que dejaron planteadas los experimentos anteriores. En las zonas en las cuales existía gran solapamiento, tanto el error de localización como la incerteza asociada a la pose fueron considerablemente bajos. Por otro lado, disminuyendo el solapamiento, se

pudo cubrir totalmente la escena casi sin perder precisión. De todas formas, se pudo apreciar que en ciertas zonas donde no existía cobertura por ninguna cámara o donde solo existían cámaras lejanas al patrón, la precisión del sistema cayó fuertemente. Sin embargo, se pudo ver que el problema principal en estos casos fue la estimación de la velocidad, que en este trabajo fue resuelta de una forma relativamente simple. En particular, se pueden mencionar algunos resultados interesantes como por ejemplo, que a  $10,016 m$ , la máxima distancia que logra captar una cámara al patrón en la escena, el error del método de localización es de  $0,011 m$ , cuando el de la instancia de WhyCon asociada a esa cámara en ese instante de tiempo es de  $1,174 m$ . Por otro lado, el promedio del error cometido por el método propuesto es de  $0,012 m$ , y de hecho casi el 90% de los errores cometidos están por debajo de  $0,01 m$ .

En resumen y como conclusión general de esta tesis, se puede decir que se obtuvo un método de localización preciso y robusto, que extiende fuertemente las capacidades originales del sistema WhyCon sobre una sola cámara. De esta forma, el sistema propuesto puede ser utilizado para la localización de robots móviles en grandes áreas, utilizando sensores de relativo bajo costo. Asimismo, controlando el solapamiento de las cámaras es posible afectar directamente la precisión de la estimación o el cubrimiento del sistema.

## 5.1. Trabajo futuro

A partir del trabajo realizado en esta tesis, y tras analizar los resultados obtenidos, se pueden identificar diversos aspectos que pueden ser mejorados en una futura instancia.

En primer lugar, en cuanto a la estimación de la velocidad, se podría probar incorporándola al estado para su estimación en forma robusta y, además, incorporar sensores que permitan observarla, tales como los encoders de un robot terrestre. Los experimentos realizados indican que esto podría mejorar considerablemente el paso de predicción del método, lo que a su vez implicaría una mejor estimación de la pose.

Por otro lado, con respecto al error que introduce el sistema WhyCon al observar el patrón en el borde de la imagen, se podría resolver este problema evitando que la instancia de WhyCon asociada a la cámara publique las estimaciones sobre el borde. Esto probablemente también mejoraría la estimación de la velocidad y por ende la estimación de la pose en dichas situaciones.

Además, en el estado del arte se mencionó un Filtro de Kalman particular, *Extended Kalman Smoother* utilizado en [19], el cual es conveniente en casos donde se quiera utilizar el método como *ground truth*. Se podría evaluar la calidad del método de localización utilizando este filtro comparado con el Filtro de Kalman Extendido.

Finalmente, sería interesante la evaluación del método propuesto en situaciones reales, no solamente en simulaciones, realizando experimentos con un sistema de múltiples cámaras observando un robot móvil que cuente con el patrón a seguir. En estos casos habría que tener en cuenta la calibración de las cámaras, que bien podría ser realizada con el sistema WhyCon o con el sistema Kalibr,

y la sincronización entre ellas, es decir deberían contar con un *clock* compartido. Otra situación que sería interesante evaluar podría ser la de realizar experimentos con robots aéreos de forma de localizar un robot moviéndose también en altura. Además, podría ser muy útil la experimentación variando la parametrización de las cámaras en cuanto a la resolución, amplitud de visión o distancia focal. Por último, se podrían evaluar escenarios con múltiples patrones para lo cual se podría trabajar con otros sistemas de localización como por ejemplo *Apriltag* para reconocer cada patrón. En este sentido, adaptar el método de localización sería muy sencillo, con proponer un nuevo modelo de error para este método en particular alcanzaría.

# Bibliografía

- [1] Andreas Breitenmoser, Laurent Kneip, and Roland Siegwart. A monocular vision-based system for 6D relative robot localization. *IEEE International Conference on Intelligent Robots and Systems*, pages 79–85, 2011.
- [2] Yutaka Yamamoto, Paolo Pirjanian, Mario Munich, Enrico DiBernardo, Luis Goncalves, Jim Ostrowski, and Niklas Karlsson. Optical sensing for robot perception and localization. *2005 IEEE Workshop on Advanced Robotics and its Social Impacts*, 2005(July 2005):14–17, 2005.
- [3] Mark Fiala. ARTag, An Improved Marker Based System Based on ARToolkit. *System*, 47166(July):36, 2004.
- [4] Daniel Wagner and Dieter Schmalstieg. ARToolKitPlus for Pose Tracking on Mobile Devices. *Proceedings of 12th Computer Vision Winter Workshop CVWW07*, pages 139–146, 2007.
- [5] Dr. Hirokazu Kato. ARToolkit.
- [6] David Meger, Ioannis Rekleitis, and Gregory Dudek. Simultaneous planning, localization, and mapping in a camera sensor network. *Distributed Autonomous Robotic Systems 7*, 54:155–164, 2006.
- [7] E Stump, V Kumar, B Grocholsky, and P M Shiroma. Control for Localization of Targets using Rangeonly Sensors. *International Journal of Robotics Research*, may 2009.
- [8] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3400–3407. IEEE, 2011.
- [9] Michael Rubenstein, Adrian Cabrera, Justin Werfel, Golnaz Habibi, James McLurkin, and Radhika Nagpal. Collective transport of complex objects by simple robots. *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, pages 47–54, 2013.
- [10] Matthias Faessler, Elias Mueggler, Karl Schwabe, and Davide Scaramuzza. A monocular pose estimation system based on infrared LEDs. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 907–913, 2014.



- [11] Andrew Richardson, Johannes Strom, and Edwin Olson. AprilCal: Assisted and repeatable camera calibration. *IEEE International Conference on Intelligent Robots and Systems*, pages 1814–1821, 2013.
- [12] Yangming Li, Shuai Li, and Yunjian Ge. A biologically inspired solution to simultaneous localization and consistent mapping in dynamic environments. *Neurocomputing*, 104:170–179, 2013.
- [13] Chen Feng and Vineet R. Kamat. Plane registration leveraged by global constraints for context-aware AEC applications. *Computer-Aided Civil and Infrastructure Engineering*, 28(5):325–343, 2013.
- [14] Felix Duvallet, Matthew R. Walter, Thomas Howard, Sachithra Hemachandra, Jean Oh, Seth Teller, Nicholas Roy, and Anthony Stentz. Inferring maps and behaviors from natural language instructions. *Springer Tracts in Advanced Robotics*, 109:373–388, 2016.
- [15] Diego López De Ipiña, Paulo R S Mendonça, and Andy Hopper. TRIP: A low-cost vision-based location system for ubiquitous computing. *Personal and Ubiquitous Computing*, 6(3):206–219, 2002.
- [16] Tomáš Krajník, Matías Nitsche, Jan Faigl, Petr Vanek, Martin Saska, Libor Preucil, Tom Duckett, and Marta Mejail. A Practical Multirobot Localization System. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 76(3-4):539–562, 2014.
- [17] Vicon. Vicon: ViconMX Systems. <http://www.vicon.com>. [cited 8 Jan 2014].
- [18] Simone Ceriani, Giulio Fontana, Alessandro Giusti, Daniele Marzorati, Matteo Matteucci, Davide Migliore, Davide Rizzi, Domenico G. Sorrenti, and Pierluigi Taddei. Rawseeds ground truth collection systems for indoor self-localization and mapping. *Autonomous Robots*, 27(4):353–371, 2009.
- [19] Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Derivation of Extended Kalman Filtering and Smoothing Equations Forward Recursions. *Matrix*, (1):1–5, 2004.
- [20] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*, volume 53. 2013.
- [21] Cristina Losada, Manuel Mazo, Sira Palazuelos, Daniel Pizarro, and Marta Marrón. Multi-camera sensor system for 3D segmentation and localization of multiple mobile robots. *Sensors*, 10(4):3261–3279, 2010.
- [22] Morgan Quigley, Ken Conley, Brian Gerkey, Josh FAust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Mg. ROS: an open-source Robot Operating System. *Icra*, 3(Figure 1):5, 2009.
- [23] Richard Hartley. *Multiple View Geometry in Computer Vision*, volume 1. 2015.

- [24] Farshad Arvin, Abdolrahman Attar, Ali Emre Turgut, and Shigang Yue. Power-Law Distribution of Long-Term Experimental Data in Swarm Robotics.
- [25] Tomáš Krajník, Jaime Pulido Fentanes, Oscar M Mozos, Tom Duckett, Johan Ekekrantz, and Marc Hanheide. Long-Term Topological Localization for Service Robots in Dynamic Environments using Spectral Maps. *Iros '14*, 2014.
- [26] Martin Saska, Vojtěch Spurn, and Vojtěch Vonásek. Predictive control and stabilization of nonholonomic formations with integrated spline-path planning.
- [27] Tomáš Krajník, Miroslav Kulich, Lenka Mudrová, Rares Ambrus, and Tom Duckett. IEEE Copyright Notice Where's Waldo at time t? Using Spatio-Temporal Models for Mobile Robot Search.
- [28] Vojtěch Vonásek, David Oertel, Sergej Neumann, and Heinz Wörn. Failure recovery for modular robot movements without reassembling modules.
- [29] Juan Jose Tarrío and Sol Pedre. Realtime edge-based visual odometry for a monocular camera. In *Proceedings of the IEEE International Conference on Computer Vision*, 2016.
- [30] Tadeo Corradi, Peter Hall, and Pejman Iravani. *Advances in Autonomous Robotics Systems*, volume 8717. 2014.
- [31] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35, 1960.
- [32] Bill Triggs, Philip Mclauchlan, Richard Hartley, and Andrew Fitzgibbon. *Bundle Adjustment - A Modern Synthesis*, volume 1883. 2000.
- [33] Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1280–1286, 2013.
- [34] Eric Rohmer, Surya P N Singh, and Marc Freese. V-REP : a Versatile and Scalable Robot Simulation Framework. pages 0–5.