UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

Departamento de Electrónica

Máster Oficial en Sistemas Electrónicos Avanzados.
Sistemas Inteligentes

Tesis de Máster

"6DOF LOCALIZACIÓN Y MAPEADO SIMULTÁNEO (SLAM) EN TIEMPO REAL MEDIANTE CÁMARA ESTÉREO MOVIDA CON LA MANO"

Pablo Fernández Alcantarilla

2008

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

Departamento de Electrónica

Máster Oficial en Sistemas Electrónicos Avanzados.
Sistemas Inteligentes

Tésis de Máster

"6DOF LOCALIZACIÓN Y MAPEADO SIMULTÁNEO (SLAM) EN TIEMPO REAL MEDIANTE CÁMARA ESTÉREO MOVIDA CON LA MANO"

Autor: Pablo Fernández Alcantarilla

Director/es: Luis Miguel Bergasa Pascual

Tribunal:

Presidente: D. Miguel Ángel Sotelo Vázquez.

Vocal 1º: D. Alfredo Gardel Vicente.

Vocal 2º: D. Luis Miguel Bergasa Pascual.

Calificación: .....

Fecha: .....

Índice general

Índice de figuras

Índice de tablas

Parte I

Resumen

Resumen

El presente trabajo de investigación tiene por objetivo el desarrollo de un método capaz de proporcionar la posición y la orientación absolutas de una cámara estéreo movida con la mano en un entorno tridimensional, considerando que el movimiento de la cámara no tiene ningún tipo de restricción, es decir, trabajamos con 6 grados de libertad (6 DOF).

Para conseguirlo, se aplica la técnica conocida como *Simultaneous Localization and Mapping* (SLAM) que nos permitirá obtener de manera simultánea tanto la posición de la cámara como el mapeado del entorno explorado. La solución propuesta en el presente trabajo, consiste en el mapeado progresivo del entorno a partir de una serie de marcas naturales. Estas marcas son obtenidas a partir de la información proporcionada por la propia cámara estéreo a localizar, siendo además, dichas marcas las que proporcionarán la información necesaria para estimar la localización/orientación requerida. En el presente trabajo se estudian dos métodos distintos de parametrización de las marcas: parametrización 3D (X,Y,Z) y parametrización inversa (distancia, ángulos de azimuth y elevación). El comportamiento dinámico del sistema se ha modelado mediante un método bayesiano denominado *Filtro de Kalman Extendido* (EKF).

Palabras Clave: *Localización y Mapeado Simultáneos (SLAM), Filtro Extendido de Kalman (EKF), 6 DOF, Detector de Características, Visión Estereoscópica, Parametrización Inversa, Warping.*

Parte II

Memoria

Capítulo 1

Introducción

El problema de la localización en robótica es un aspecto clave a la hora de conseguir robots verdaderamente autónomos capaces de navegar por un determinado entorno. Para que un móvil pueda localizarse, es necesario que este disponga de unas referencias de medidas, que pueden ser tanto relativas como absolutas, y que le proporcionen una realimentación de las acciones o movimientos del móvil y de su posición con respecto al entorno que le rodea. La localización y el mapeado del entorno son dos procesos dependientes que se calculan mediante el móvil navega por el entorno. Respecto al problema de localización, el problema a resolver es el denominado como *Simultaneous Localization and Mapping* (SLAM). Los sistemas basados en SLAM permiten diseñar mapas del entorno de forma automática para que posteriormente puedan ser utilizados para la navegación autónoma de robots.

En la literatura se pueden encontrar numerosas soluciones al problema de SLAM en robótica [1] [2]. Uno de los mayores problemas a la hora de localizar un móvil en un entorno tiene que ver con los **errores** cometidos al realizar las medidas con los diversos sensores que se disponga. Como consecuencia de estos errores, existe una **incertidumbre** asociada a la localización del móvil. Para tratar de resolver este problema, se han desarrollado diversos métodos probabilísticos, la mayoría de los cuáles se encuadran dentro de los denominados *Filtros Bayesianos*. Estos métodos tienen su base en la idea de que no se puede conocer con total certeza, tanto el estado del móvil (posición y orientación) como los resultados de las medidas realizadas. Por el contrario, se incorpora el concepto de incertidumbre del estado del móvil, definiendo una distribución de creencia de dicho estado. El objetivo de estos filtros bayesianos, consiste en estimar el estado del sistema en cada instante de tiempo, primero realizando una predicción de dicho estado y posteriormente tras realizar medidas se corrige y se actualiza dicha predicción. Entre los ejemplos de métodos bayesianos se pueden citar el *Filtro de Kalman* (KF), *Filtro Extendido de Kalman* (EKF), *Filtros de Partículas* (PF) [3] [4] [5].

A la hora de representar el mapa del entorno, existen fundamentalmente dos posibilidades: el **enfoque topólogico** y el **enfoque métrico**. El primero se basa en definir un número discreto de estados, definido por diversas características (al lado de una puerta, entre dos pasillos, a la altura de una mesa, etc.). Por el contrario, el enfoque métrico se basa en que el estado representa la posición real del objeto respecto del entorno, es decir, el estado puede tomar cualquier valor.

Hasta hace poco tiempo, la mayoría de los sistemas SLAM usaban principalmente sensores scan-láser y estaban centrados en la construcción de mapas exclusivamente 2D, consiguiendo buenos resultados para robots controlados y aplicando modelos dinámicos precisos [6] [7]. Sin embargo, en los últimos años recientes investigaciones han demostrado una gran utilidad de los métodos de SLAM basados en cámaras (tanto monoculares como estéreo) para aplicaciones en

las que el objetivo es obtener en tiempo real la posición 3D de una cámara, la cuál puede moverse rápidamente en un entorno determinado. Dichos métodos se basan en el mapeado de una serie de marcas visuales dispersas por el entorno, evitando además la necesidad de un conocimiento detallado de la dinámica de movimiento [4].

Muy reciéntemente el uso de sistemas basados en SLAM Visuales está sobrepasando los límites de las aplicaciones robóticas, introduciéndose en campos como la localización de personas [8] [9] o de soporte a la cirugía mínimamente invasiva [10].

El presente trabajo de investigación tutelado se propone como un Sistema de Localización y Posicionamiento Simultáneos mediante visión artificial para asistir a la navegación de las personas invidentes. Como primer paso se ha desarrollado un sistema basado en un SLAM métrico inspirado en el trabajo previo de D. Schleicher et al. [11]. Sin embargo, en este trabajo se debe de abordar la problemática de una cámara estéreo gran angular movida con la mano y con 6 grados de libertad. A su vez se ha realizado un estudio comparando distintos métodos de extracción de características , se ha incorporado al sistema la parametrización inversa para poder mejorar la estimación de la posición de las marcas lejanas (debido a los errores intrínsecos de la reconstrucción 3D estereoscópica), y se han estudiado diversos métodos de adaptación de parches.

El resto del documento queda estructurado de la siguiente manera:

- **Estado del Arte**: En esta sección, se comentan brevemente los principales trabajos de investigación en las líneas de localización de personas y/o asistencia a la navegación de personas invidentes utilizando como sensores principales cámaras.
- **Sistema Físico**: En esta sección, se describen las características del sensor utilizado, así cómo la configuración del sistema físico.
- **Visión Estereoscópica**: En esta sección, se comentan los principales fundamentos teóricos de la visión estéreo, que posteriormente serán utilizados en el algoritmo de Visual SLAM.
- **Extracción de Características**: El objetivo final del sistema, es la obtención de un sistema de localización métrico mediante el mapeado secuencial en 3D de una serie de marcas naturales extraídas del entorno. Por lo tanto, en esta sección se detallan diversos algoritmos de extracción de características de *bajo nivel* que se han utilizado en el desarrollo del trabajo.
- **Visual SLAM**: En esta sección se describe detalladamente el algoritmo de SLAM implementado, dedicando especial atención al Filtro Extendido de Kalman (EKF), con sus fases de predicción y actualización y al modelo de movimiento implementado. También se detallan los dos tipos de parametrizaciones de marcas implementadas: parametrización 3D y parametrización inversa.
- **Resultados**: En esta sección, se muestran los resultados experimentales del sistema, así como diversas comparativas entre los distintos métodos empleados.

Capítulo 2

Estado del Arte

Hasta hace pocos años, las técnicas SLAM se utilizaban básicamente en el campo de la robótica móvil utilizando principalmente sensores scan-láser. A medida, que la investigación en visión artificial fue avanzando, se demostró que el uso de sistemas de visión para realizar localización y mapeado es posible. Incluso, la aplicación de estas técnicas está sobrepasando la frontera de las aplicaciones robóticas, aplicándose a campos tan dispares como la localización de personas o la cirugía mínimamente invasiva.

Las técnicas SLAM suelen presentar problemas de convergencia y tiempo de ejecución sobre grandes entornos. Para conseguir resultados precisos, se necesitan utilizar algoritmos robustos complejos, que provocan un coste computacional (añadido al coste computacional del propio SLAM) lo suficientemente elevado como para impedir su funcionamiento en tiempo real. Además, como el objetivo final es un sistema de asistencia a personas invidentes, hay que tener en cuenta que esta *plataforma* no se puede detener a voluntad.

En esta sección, se detallan los principales trabajos de investigación en la línea de localización de personas. En principio, cualquier trabajo que tenga como objetivo la localización de personas, es un punto de partida y una base sólida para un sistema de asistencia a la navegación de invidentes. Aunque no obstante, los sistemas de navegación de invidentes deben de cumplir unos requisitos especiales para su uso por personas invidentes, y lo que es más importante es necesaría una realimentación por parte de las personas invidentes de las características *realmente* necesarias que debe presentar un sistema de estas características.

Dentro de este tipo de sistemas, existen grupos de investigación que trabajan únicamente con una sola cámara, otros grupos que trabajan con visión estéreo y/o fusión de otros sensores como GPS, o información de marcas sonoras por el entorno. A continuación se detallan los principales trabajos en el área de investigación haciendo especial hincapie en el cmapo de SLAM métrico.

2.1. MonoSLAM

2.1.1. Universidad de Zaragoza, Imperial College of London

La investigación en el campo de técnicas SLAM utilizando únicamente como sensor visión (ya sea mono o estéreo), tiene uno de sus principales referentes en la figura de Andrew J. Davison, profesor del Imperial College of London, que fue de los pioneros en aplicar Visual SLAM a un sistema robótico basado en una cámara estéreo [12]. Sin embargo, con el paso de los años Andrew J. Davison apostó por la visión monocular en contra de la visión estereoscópica desarrollando

un gran trabajo en este campo. A su vez, la colaboración de Andrew J. Davison con varios investigadores de la Universidad de Zaragoza, y con antiguos colaboradores de Oxford University, han supuesto resultados muy fructíferos, convirtiéndose en grandes referencias mundiales en este campo.

El trabajo [13] presenta un SLAM Monocular en el cuál la novedad, es que la cámara es movida con la mano. De tal modo que lo único necesario es una cámara de bajo coste, un cable firewire y un ordenador portátil, como se puede apreciar en la figura 2.1.

Figura 2.1: Sistema Monocular movido con la mano. Universidad de Zaragoza

Este sistema utiliza el EKF-SLAM para consturir mapas locales e independientes haciendo uso de la parametrización inversa de las marcas, para posteriormente relacionarlos en un proceso de mapeado jerárquico de alto nivel. Además, para evitar posibles emparejamientos erróneos de marcas, utiliza el algoritmo denomindado como *Joint Compatibility Test* [14] logrando reducir el número de *outliers* o marcas erróneas.

Para la extracción de marcas o puntos de interés, utilizan el operador de Shi-Tomasi [15]. La búsqueda de las marcas en los siguientes frames se realiza por medio de una correlación normalizada del parche original en el área de incertidumbre o de búsqueda de la marca en los sucesivos frames.

Uno de los problemas de trabajar con visión monocular, es que solamente es posible obtener de una manera directa dos de las tres coordenadas de la posición relativa de las marcas, pero sin embargo, no es posible conocer la coordenada de profundidad de dichas marcas. Para solventar dicho problema, se pueden elegir dos alternativas:

- Partir de marcas conocidas a priori, utilizando algún patrón determinado, para poder obtener la profundidad de dichas marcas. Sin embargo, cada vez que se quieren añadir nuevas marcas al mapa es necesario esperar unos cuantos frames hasta lograr reducir la incertidumbre asociada a la profundidad de la marca, antes de incorporarla al filtro EKF.
- Utilizando la parametrización inversa de las marcas propuesta en [16] se consigue una inicialización directa de las marcas, en términos de ángulos de azimuth y de elevación, y de la inversa de la profundidad, obteniendo ecuaciones de medida con un alto grado de linealidad.

Los principales problemas que presenta este sistema, es que no es capaz de funcionar en tiempo real, y que presenta problemas de convergencia sobre grandes entornos. Además, al no

realizar ningún método de adaptación de parches, los cambios en el punto de vista de la cámara deben de ser pequeños, y los movimientos por lo tanto deben ser bastante controlados.

2.2. StereoSLAM

2.2.1. Universidad de Zaragoza

Este trabajo desarrollado por un equipo de investigadores de la Universidad de Zaragoza, utiliza conocimientos similares al trabajo anteriormente descrito basado en MonoSLAM, con la diferencia, de que utilizan información de un par estéreo comercial, realizando un EKF-SLAM de 6 grados de libertad [8]. En la figura 2.2 se puede ver un ejemplo del sistema:

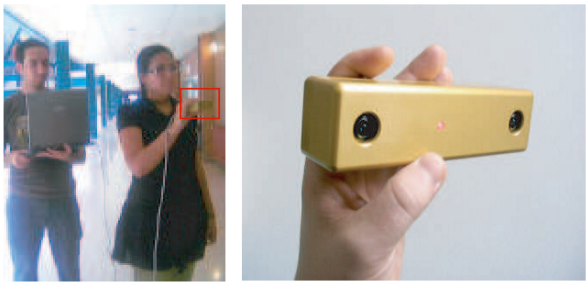


Figura 2.2: Sistema Estéreo movido con la mano. Universidad de Zaragoza

El sistema también utiliza la parametrización inversa para marcas lejanas. Aunque con la información estéreo se conoce inmediatamente la profundidad de una marca, la parametrización inversa proporciona una mejor representación angular para marcas lejanas, que una parametrización basada en 3D, en dónde los errores propios del estéreo para marcas lejanas pueden ser considerables.

Los principales problemas que presenta este sistema son similares a los descritos en el sistema MonoSLAM. El sistema no es capaz de funcionar en tiempo real, presenta problemas de convergencia sobre grandes entornos, y tampoco se realiza ningún método de adaptación de parches, por lo que los cambios en el punto de vista de la cámara deben de ser pequeños, y los movimientos controlados.

2.2.2. Universidad de Alicante

El trabajo desarrollado por los investigadores de la Universidad de Alicante, constituye una de las primeras aproximaciones en el campo de la asistencia a la navegación para personas invidentes [9]. Para poder obtener un SLAM 6DOF, construyeron un prototipo *wearable* como se muestra en la figura 2.3. En esta figura se observa como el prototipo consta únicamente de la cámara estéreo y un PC portátil de pequeñas dimensiones. El sistema se transporta con una mochila cruzada, fijando la cámara en la parte delantera y el PC portátil en la bolsa trasera.

Este trabajo se diferencia de los anteriores, en que sigue una metodología totalmente distinta a la de los estos trabajos, inspirados fundamentalmente en las investigaciones de Andrew J. Davison.

A partir de un sistema estéreo y utilizando algortimos de Visual SLAM considerando 6DOF,

Figura 2.3: Sistema Estéreo. Universidad de Alicante

se realiza una estimación del propio movimiento o *egomotion* mediante algoritmos de matching 3D, y un mapeado a través de algoritmos de minimización de entropía global. La aplicación de este sistema está limitada a escenarios de interiores y ortogonales (como pasillos, corredores...), siendo difícil su extensión a otros escenarios más complejos y no ortogonales. Otra problemática añadida a este trabajo, es que al realizar un SLAM denso, no es capaz de funcionar en tiempo real.

2.3. System for Wearable Audio Navigation

El sistema SWAN (*System for Wearable Audio Navigation*) ha sido desarrollado por los investigadores del prestigioso Georgia Institute of Technology destacando Frank Dellaert, especialista en robótica móvil y Bruce Walker, especialista en audio [17]. El prototipo actual del sistema SWAN consiste en un pequeño ordenador portatil transportado en una mochila, 4 cámaras, un sensor de luz, y unos auriculares especiales. La información se transmite a la persona invidente mediante señales de audio, ya que las personas invidentes dependen fundamentalmente de su sentido del oido, más desarrollado que en el resto de personas. Un ejemplo de este sistema se puede ver en la figura 2.4.

Además de los sensores ya citados, utilizan también la información procedente de un GPS y tecnología RFID (Radio Frequency IDentification o Identificación por Radiofrecuencia). Mediante una fusión sofisticada de la información procedente de los distintos sensores, logran determinar la localización de la persona y el camino en el que se encuentra. Una vez que la localización ha sido determinada, SWAN utiliza una interfaz de audio (básicamente una serie de sonidos puntuales denominados *beacons*) para guiar al usuario a lo largo del entorno. Uno de los aspectos más interesantes del proyecto, es la utilización de una plataforma con 4 cámaras que permite la localización en un mapa 3D conocido del entorno. Para obtener la localización, utilizan la información de las cámaras y posteriormente aplican una estimación de la localización utilizando

Figura 2.4: System Wearable Audio Navigation

RANSAC, de tal modo que no realizan ninguna correspondencia o *matching* entre el mapa 3D y las características extraídas de las imágenes.

De momento el sistema solamente se encuentra funcionando en exteriores. El principal problema existe en interiores, en dónde se debe refinar mucho el sistema basado en visión para poder obtener buenos mapas del entorno. Otro de los principales problemas, es el uso de numerosos sensores, muchos de ellos los cuáles no son necesarios, como por ejemplo, el uso de 4 cámaras. Sin embargo, una de las cosas positivas con las que cuenta este sistema, y que por el momento las otras investigaciones no disponen, es de la realimentación de las propias personas invidentes, gracias a la colaboración en el proyecto del *Center for the Visually Impaired in Atlanta*.

Capítulo 3

Sistema Físico

El sensor empleado consiste en dos cámaras estéreo *Unibrain Fire-i* de óptica gran angular. Las características que presentan dichas cámaras son las siguientes:

Interfaz	IEEE-1394a FireWire 400 Mbps, 2 puertos, 6 pines
Resolución	VGA 640x480
Óptica	Gran Angular (160' H, 116' V)
Formatos de Imagen	YUV (4:1:1, 4:2:2, 4:4:4), RGB-24 bit, Monocromo 8 bit
Frame Rate	30, 15, 7.5, 3.75 fps
Alimentación	8 a 30 VDC, por bus 1394 o jack externo. Consumo 1W max, 0.9W típico, 0.4W modo sleep

Tabla 3.1: Características Cámaras Unibrain Fire-i

Figura 3.1: Aspecto Cámaras Unibrain Fire-i

Durante pruebas experimentales, se comprobó que las cámaras Unibrain Fire-i, presentan una desincronización considerable que aumenta con el tiempo. Esta desincronización no es posible corregirse mediante software y/o hardware con la configuración de cámaras actuales. En un sistema estéreo, una desincronización elevada puede llevar a graves consecuencias, como por ejemplo, que no se encuentren correspondencias entre las dos imágenes en un instante determinado de tiempo. Para mitigar el efecto de esta desincronización, se utiliza un Hub FireWire 400

Mbps 6-Puertos como el que se observa en la figura 3.2.

Figura 3.2: Hub Firewire 400 6-Puertos

En la figura 3.3 se puede ver una prueba con el sistema físico utilizado para este trabajo, en dónde se pueden apreciar, el par estéreo movido con la mano, los cables firewire, el hub y el ordenador portátil.

Figura 3.3: Prueba Experimental: Sistema Estéreo, Cables Firewire y Ordenador Portátil

Capítulo 4

Visión Estereoscópica

La visión estereóscopica consiste en la utilización de dos cámaras (par estéreo) para extraer información tridimensional del entorno. Es decir, tiene por objetivo realizar una reconstrucción de la estructura tridimensional (3D) del espacio a partir de la adquisición simultánea de dos imágenes. Conociendo el modelo de proyección de cada cámara y la relación espacial entre ellas, se puede calcular las coordenadas 3D de un punto a partir de su proyección en las dos imágenes. A continuación se va exponer el modelo de cámara utilizado y la teoría asociada con la visión estéreo.

4.1. Modelado de la cámara

La óptica geométrica clásica se basa en los modelos de lente gruesa y lente fina. Partimos del modelo *pin-hole* [18] donde todos los rayos pasan por un único punto, denominado centro óptico (figura 4.1). La matriz CCD (plano imagen) se encuentra a una distancia f del centro óptico, distancia denominada *distancia focal*.

Se puede observar que la escena al proyectarse en el plano imagen se invierte. Para resolver este problema se sitúa un plano imagen equivalente por delante del centro óptico a la misma distancia f. Desde un punto de vista físico, esto se realiza leyendo la matriz CCD de forma que se obtenga la imagen inversa (figura 4.2).

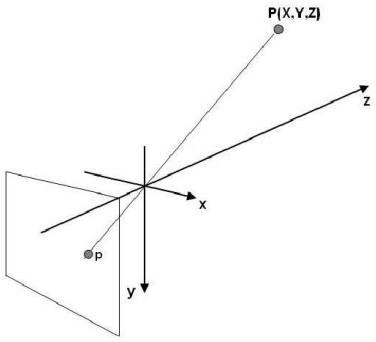


Figura 4.1: Geometría de un sistema de formación de imágenes

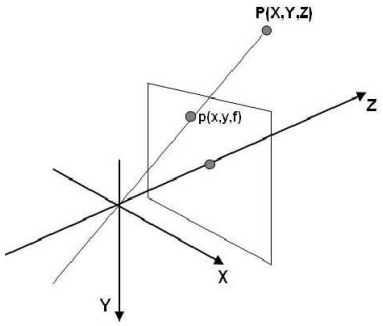


Figura 4.2: Plano imagen definido delante del centro óptico

4.1.1. Formación de imágenes: proyección perspectiva

La proyección perspectiva consiste en una abstracción geométrica en la que la proyección de un punto sobre el plano imagen viene dada por el punto de corte de la línea visual que une el punto 3D y el centro óptico de la cámara con el plano imagen.

A partir de ahora se utilizará la siguiente notación:

- (X_c, Y_c, Z_c): sistema de coodernadas (referencia) de la cámara
- (X_w, Y_w, Z_w): sistema de coordenadas ligado a la modelización del objeto
- (x, y, z): sistema de cordenadas de la imagen (sin discretizar)
- (u, v): sistema de coordenadas discreto de la imagen

4.1.2. Cambio de referencia Objeto/Cámara

El primer cambio de referencia permite expresar las coordenadas del patrón de calibración (posición en el espacio) referenciadas a la cámara, con una rotación y una traslación.

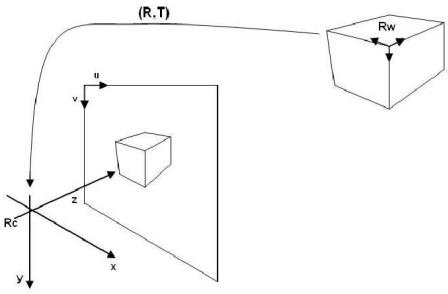


Figura 4.3: Diferentes referencias necesarias para la calibración

$$\mathbf{P_c} = (\mathbf{M_1}) \cdot \mathbf{P_w} \tag{4.1}$$

$$\mathbf{P_c} = \begin{pmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \tag{4.2}$$

donde P_w son las coordenadas 3D de un punto del patrón expresado bajo la referencia del modelo y P_c son las coordenadas del mismo punto expresado en la referencia de la cámara.

Las matrices de rotación y traslación ($\mathbf{R_{(3x3)}}$ y $\mathbf{T_{(3x1)}}$) se definen bajo el sistema de referencia de la cámara.

4.1.3. Cambio de referencia Cámara/Imagen

El cambio de referencia de la cámara a la imagen está ligado a las ecuaciones de proyección perspectiva. Teniendo en cuenta la figura 4.4, estas ecuaciones se expresan de la siguiente forma:

$$\begin{gathered} x = X_c \cdot f/Z_c \\ y = Y_c \cdot f/Z_c \\ z = f \end{gathered} \tag{4.3}$$

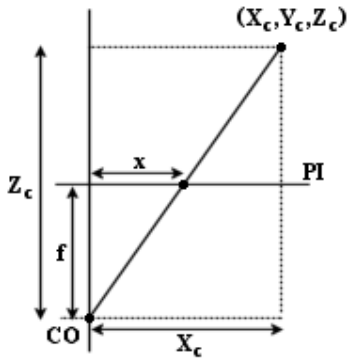


Figura 4.4: Cambio de coordenadas Cámara/Imagen

En coordenadas homogéneas el sistema se escribe:

$$\begin{pmatrix} sx \\ sy \\ s \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \mathbf{P_c} \tag{4.4}$$

La notación homogénea introduce un factor multiplicativo s, en el paso $(\Re^3, \Re^2)$:

$$\begin{cases} sx = f \cdot X_c \\ sy = f \cdot Y_c \\ s = Z_c \end{cases} \tag{4.5}$$

sustituyendo s:

$$\begin{cases} x = f \cdot X_c/Z_c \\ y = f \cdot Y_c/Z_c \end{cases} \tag{4.6}$$

4.1.4. Cambio de coordenadas en el plano imagen

Se trata de expresar las coordenadas x e y relativas a la forma de un píxel elemental de la matriz CCD. En concreto la digitalización consiste en un cambio de unidades (de mm a píxeles) y una traslación (de la referencia de la cámara al inicio del plano imagen -esquina superior izquierda-).

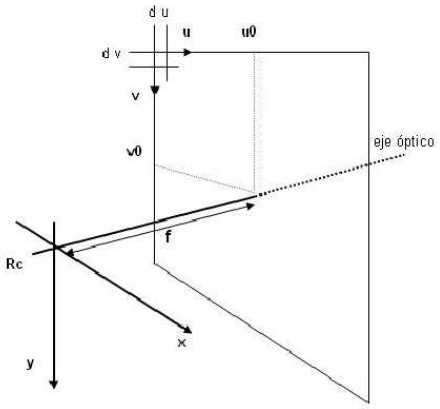


Figura 4.5: Sistema de coordenadas en la matriz CCD

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} 1/dx & 0 & u_0 \\ 0 & 1/dy & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} sx \\ sy \\ s \end{pmatrix} \tag{4.7}$$

Resolviendo tenemos las ecuaciones clásicas de cambio de referencia (coordenadas de la cámara/coordenadas de píxel):

$$\begin{cases} u &= x/dx + u_0 \\ v &= y/dy + v_0 \end{cases} \tag{4.8}$$

donde (u_0, v_0) representan las coordenadas (en píxeles) en la imagen de la intersección del eje óptico y el plano imagen y (dx, dy) son las dimensiones (por ejemplo, en mm/píxel) en x e y de un píxel elementario de la matriz CCD de la cámara.

4.1.5. Expresión general

El sistema completo de formación de la imagen se expresa según la relación siguiente:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} 1/dx & 0 & u_0 \\ 0 & 1/dy & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \tag{4.9}$$

Tenemos los siguientes parámetros:

- (X_w, Y_w, Z_w) son las coordenadas 3D de un punto del patrón
- (u, v) son las coordenadas 2D del píxel en la imagen de la proyección del punto anterior

- (u_0, v_0, f, dx, dy) son los llamados **parámetros intrínsecos** de calibración que son propios del sistema de adquisición.
- $(r_{(11,\ldots,33)}, t_{(x,y,z)})$ son los llamados **parámetros extrínsecos** de calibración.

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \mathbf{M_{int}}\mathbf{M_{ext}} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = \mathbf{M_{(3x4)}} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}, \tag{4.10}$$

donde $\mathbf{M}$ es la matriz de calibración del sistema que agrupa los parámetros intrínsecos y extrínsecos. Normalmente la matriz de parámetros intrínsecos se expresa de la siguiente forma:

$$\mathbf{M_{int}} = \begin{pmatrix} f/dx & 0 & u_0 & 0 \\ 0 & f/dy & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{4.11}$$

Luego:

$$\left.\begin{array}{l} f_x = f/dx \\ f_y = f/dy \end{array}\right\} \rightarrow (f_x/f_y = d_y/d_x). \tag{4.12}$$

La relación d_x/d_y representa la relación pixélica. Los parámetros intrínsecos que se calculan en un proceso de calibración, por tanto son 4, (f_x, f_y, u_0, v_0), mientras que los extrínsecos son 12; 9 para la rotación, $(r_{11}, \ldots, r_{33})$ y 3 para la traslación, $(t_{(x,y,z)})$, que son independientes de la cámara.

En definitiva, cada elemento de la matriz de calibración $\mathbf{M}$ se calcula como:

$$\begin{cases} m_{1(1,2,3)} = f_x \cdot r_{1(1,2,3)} + u_0 \cdot r_{3(1,2,3)} \\ m_{14} = f_x \cdot t_x + u_0 \cdot t_z \\ m_{2(1,2,3)} = f_y \cdot r_{2(1,2,3)} + v_0 \cdot r_{3(1,2,3)} \\ m_{24} = f_y \cdot t_y + v_0 \cdot t_z \\ m_{3(1,2,3)} = r_{3(1,2,3)} \\ m_{34} = t_z \end{cases} \tag{4.13}$$

4.2. Detección estéreo

La calibración de una sola cámara no nos permite recuperar la información tridimensional de un entorno a través de las imágenes que de él captamos. Esto se debe a que por cada punto 2D en el plano imagen, se obtienen dos ecuaciones de proyección independientes (ec. (4.9)), mientras son 3 las incógnitas a resolver (coordenadas tridimensionales del punto).

Sin embargo, muchas aplicaciones basadas en visión, como son la modelización de objetos, la navegación de vehículos (que abarca a la odometría visual) y la inspección geométrica, requieren información tridimensional, tanto métrica como no métrica. Una de las soluciones es el uso de múltiples vistas [19], pero esto lleva a que aparezcan nuevos problemas. El primero es que tendremos que encontrar de nuevo una relación entre las imágenes que captamos con las cámaras y el mundo 3D. El segundo es que para que la información "redundante" que captamos con la segunda cámara resulte de alguna utilidad debemos conocer qué puntos en ambas imágenes corresponden al mismo punto físico de la escena para así poder utilizar esa información para calcular la profundidad.

4.2.1. Planteamiento del problema

Comenzaremos abordando el primer problema. Sea un sistema estereoscópico como el de la figura 4.6.

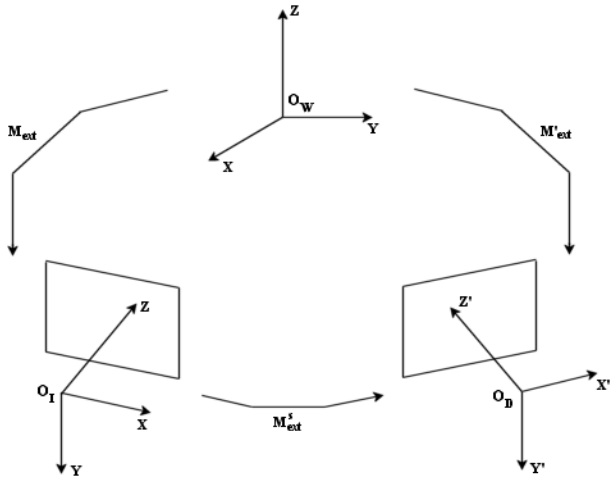


Figura 4.6: Sistema estereoscópico

El objetivo ahora es calcular la relación que existe entre el mismo punto en las dos imágenes y el punto físico 3D. Para ello, igual que en el caso de una sola cámara, se deben realizar una serie de transformaciones y cambios de sistema de referencia hasta encontrar la relación entre los tres puntos.

4.2.2. Calibración estereoscópica

En este proceso de calibración, calcularemos la relación entre un punto (X_w, Y_w, Z_w) con coordenadas en el sistema de referencia de la cámara derecha (X', Y', Z') y el mismo punto expresado en el sistema de referencia de la cámara izquierda (X, Y, Z). Es decir, se calcula la

matriz de transformación entre los sistemas de coordenadas de la cámara izquierda y de la cámara derecha. Para ello suponemos que previamente hemos seguido los siguientes pasos:

1. Calibrar cada cámara a partir de un sistema de coordenadas único. Esto nos proporciona las matrices de parámetros intrínsecos y extrínsecos $\mathbf{M}$ y $\mathbf{M'}$.
2. Extraer los parámetros intrínsecos y extrínsecos de cada cámara, $\mathbf{M_{int}}$, $\mathbf{M_{ext}}$ y $\mathbf{M'_{int}}$, $\mathbf{M'_{ext}}$.

Con estos datos es fácil calcular la matriz de transformación entre el sistema de coordenadas de la cámara izquierda y el sistema de coordenadas de la cámara derecha.

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{M_{ext}} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}; \begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} = \mathbf{M'_{ext}} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \tag{4.14}$$

$$\mathbf{M_{ext}^{-1}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} = \mathbf{M'_{ext}} \mathbf{M_{ext}^{-1}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{4.15}$$

$$\mathbf{M_{ext}^{s}} = \mathbf{M'_{ext}} \mathbf{M_{ext}^{-1}} \tag{4.16}$$

Hemos obtenido la matriz que relaciona los sistemas de las dos cámaras:

$$\mathbf{M_{ext}^{s}} = \begin{pmatrix} r_{11}^s & r_{12}^s & r_{13}^s & t_x^s \\ r_{21}^s & r_{22}^s & r_{23}^s & t_y^s \\ r_{31}^s & r_{32}^s & r_{33}^s & t_z^s \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{r_1^s} & t_x^s \\ \mathbf{r_2^s} & t_y^s \\ \mathbf{r_3^s} & t_z^s \\ \mathbf{0} & 1 \end{pmatrix} \tag{4.17}$$

donde $\mathbf{T^s} = (t_x^s, t_y^s, t_z^s)^t$ es el vector que va de F a F'.

4.2.3. Reconstrucción tridimensional

La reconstrucción tridimensional a partir de un par de imágenes presupone que existe correspondencia entre ambas, es decir, que se conocen todos los pares formados por un punto de la imagen izquierda y otro punto de la imagen derecha que son la representación del mismo objeto físico captado por ambas cámaras. Posteriormente veremos cómo hallar esa correspondencia (apartado 4.2.8).

Partimos de que tenemos nuestras cámaras calibradas individualmente y es conocida la transformación entre sus sistemas de coordenadas (calibración estéreo) calculada haciendo uso de la ecuación 4.16. Tomando la relación entre un punto 3D y su proyección en la imagen que nos da la ecuación 4.9 y escribiendo esa relación para un punto 3D (X_w, Y_w, Z_w) sobre la imagen izquierda (u, v) y sobre la derecha (u', v') tenemos dos pares de ecuaciones:

$$\begin{cases} u = \frac{m_{11} \cdot X_w + m_{12} \cdot Y_w + m_{13} \cdot Z_w + m_{14}}{m_{31} \cdot X_w + m_{32} \cdot Y_w + m_{33} \cdot Z_w + m_{34}} \\ v = \frac{m_{21} \cdot X_w + m_{22} \cdot Y_w + m_{23} \cdot Z_w + m_{24}}{m_{31} \cdot X_w + m_{32} \cdot Y_w + m_{33} \cdot Z_w + m_{34}} \end{cases} \tag{4.18}$$

$$\begin{cases} u' &= \frac{m'_{11}\cdot X_w + m'_{12}\cdot Y_w + m'_{13}\cdot Z_w + m'_{14}}{m'_{31}\cdot X_w + m'_{32}\cdot Y_w + m'_{33}\cdot Z_w + m'_{34}} \\ v' &= \frac{m'_{21}\cdot X_w + m'_{22}\cdot Y_w + m'_{23}\cdot Z_w + m'_{24}}{m'_{31}\cdot X_w + m'_{32}\cdot Y_w + m'_{33}\cdot Z_w + m'_{34}} \end{cases} \tag{4.19}$$

Combinando ambos sistemas llegamos a un único sistema con 4 ecuaciones y 3 incógnitas (las coordenadas 3D del punto en cuestión):

$$\begin{cases} (u\cdot m_{31} - m_{11})X_w + (u\cdot m_{32} - m_{12})Y_w + (u\cdot m_{33} - m_{13})Z_w &= m_{14} - u\cdot m_{34} \\ (v\cdot m_{31} - m_{21})X_w + (v\cdot m_{32} - m_{22})Y_w + (v\cdot m_{33} - m_{23})Z_w &= m_{24} - v\cdot m_{34} \\ (u'\cdot m'_{31} - m'_{11})X_w + (u'\cdot m'_{32} - m'_{12})Y_w + (u'\cdot m'_{33} - m'_{13})Z_w &= m'_{14} - u'\cdot m'_{34} \\ (v'\cdot m'_{31} - m'_{21})X_w + (v'\cdot m'_{32} - m'_{22})Y_w + (v'\cdot m'_{33} - m'_{23})Z_w &= m'_{24} - v'\cdot m'_{34} \end{cases} \tag{4.20}$$

Sistema que se puede escribir en forma matricial como:

$$\mathbf{A}\cdot\mathbf{P_w} = \mathbf{b} \tag{4.21}$$

Teniendo en cuenta que A no es una matriz cuadrada, para resolver calculamos su pseudo-inversa obteniendo:

$$\mathbf{P_w} = \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} = (\mathbf{A^t A})^{-1}\cdot\mathbf{A^t b} \tag{4.22}$$

A través de este cálculo y a partir de los puntos (u, v), (u', v') (proyecciones 2D) podemos obtener la **posición 3D de un punto** captado por las dos cámaras. El problema es que en la mayoría de las aplicaciones, la correspondencia entre los puntos de las imágenes no se conoce (como en nuestro proyecto). Es decir, a priori, no disponemos del par (u, v), (u', v') correspondientes a un punto físico (X_w, Y_w, Z_w). Partiendo de las imágenes se deberá deducir esa correspondencia mediante el proceso de emparejamiento de puntos o *matching* descrito posteriormente.

4.2.4. La Geometría Epipolar

Un número considerable de investigadores han estudiado cómo se puede reconocer tridimensionalmente el entorno a partir del análisis de sus imágenes 2D. Este estudio ha dado lugar a la geometría epipolar, que es una extensión de las reglas de la geometría perspectiva que nos permite conocer la relación entre un mismo punto detectado en dos imágenes distintas de la misma escena y sus coordenadas en el espacio [20].

Podemos expresar un punto **P** de la escena 3D al mismo tiempo en los dos sistemas de coordenadas. Siendo (X, Y, Z) sus coordenadas en el sistema de la cámara izquierda y (X', Y', Z') sus coordenadas en el sistema de la cámara derecha, la relación entre los dos sistemas de coordenadas viene dada en la ecuación 4.16; donde despejando obtenemos:

$$\begin{cases} X' = r^s_{11}X + r^s_{12}Y + r^s_{13}Z + t^s_x \\ Y' = r^s_{21}X + r^s_{22}Y + r^s_{23}Z + t^s_y \\ Z' = r^s_{31}X + r^s_{32}Y + r^s_{33}Z + t^s_z \end{cases} \tag{4.23}$$

4.2.5. Relación Izquierda→Derecha

Se desea establecer una relación simple entre un punto de la imagen de la izquierda y un punto de la imagen derecha. Esta relación, como se demuestra a continuación, viene dada por la

matriz esencial (para coordenadas de la cámara) o por la matriz fundamental (para coordenadas discretizadas en la imagen). En la práctica, puesto que disponemos de una matriz de puntos, se utiliza únicamente la matriz fundamental.

Para este desarrollo matemático es interesante realizar una ligera modificación en el proceso de formación de imágenes descrito en el apartado 4.1. En este caso la proyección 3D-2D se realiza sobre un plano imagen *normalizado* colocado a una distancia focal $f = 1$, en lugar de llevarse a cabo sobre el plano imagen original. En el siguiente paso (cambio de coordenadas en el plano imagen) se incluye como factor de escala la verdadera distancia focal que caracteriza a la cámara. Es decir, en este caso se utilizan las ecuaciones (4.24) y (4.25) en lugar de las ecuaciones (4.4) y (4.7).

$$\begin{pmatrix} sx \\ sy \\ s \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \tag{4.24}$$

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} f/dx & 0 & u_0 \\ 0 & f/dy & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} sx \\ sy \\ s \end{pmatrix} \tag{4.25}$$

Como vemos, simplemente se ha trasladado la inclusión de la distancia focal al último paso de la formación de la imagen. Luego, la expresión final del modelo de proyección pin-hole es idéntico al expresado por la ecuación 4.9.

Si vemos ahora cómo se proyectan esos puntos sobre el plano imagen normalizado tenemos que siendo las coordenadas de $\mathbf{p} = (x, y, z)$ (proyección de $\mathbf{P} = (X, Y, Z)$ en la imagen izquierda normalizada) donde $x = X/Z$, $y = Y/Z$ y $z = 1$ por la ecuación 4.3 y de igual manera para la imagen derecha podemos escribir la ecuación 4.23 como:

$$\begin{cases} X' = Z' \cdot x' = r^s_{11} Z \cdot x + r^s_{12} Z \cdot y + r^s_{13} \cdot Z + t^s_x \\ Y' = Z' \cdot x' = r^s_{21} Z \cdot x + r^s_{22} Z \cdot y + r^s_{23} \cdot Z + t^s_y \\ Z' = r^s_{31} Z \cdot x + r^s_{32} Z \cdot y + r^s_{33} \cdot Z + t^s_z \end{cases} \tag{4.26}$$

sustituyendo Z' y con la notación:

$$\begin{array}{ll} \mathbf{r^s_1} & = (r^s_{11}, r^s_{12}, r^s_{13}) \\ \mathbf{r^s_2} & = (r^s_{21}, r^s_{22}, r^s_{23}) \\ \mathbf{r^s_3} & = (r^s_{31}, r^s_{32}, r^s_{33}) \\ \mathbf{p} & = (x, y, 1)^t \end{array} \tag{4.27}$$

podemos despejar x' e y':

$$\begin{cases} x' = \frac{Z \cdot r^s_1 \cdot \mathbf{p} + t^s_x}{Z r^s_3 \cdot \mathbf{p} + t^s_z} \\ y' = \frac{Z \cdot r^s_2 \cdot \mathbf{p} + t^s_y}{Z r^s_3 \cdot \mathbf{p} + t^s_z} \end{cases} \tag{4.28}$$

Al eliminar Z de las ecuaciones obtenemos una relación entre un punto en la imagen izquierda y sus correspondientes en la derecha:

$$(t^s_z \cdot \mathbf{r^s_2} \cdot \mathbf{p} - t^s_y \cdot \mathbf{r^s_3} \cdot \mathbf{p}) x' + (t^s_x \cdot \mathbf{r^s_3} \cdot \mathbf{p} - t^s_z \cdot \mathbf{r^s_1} \cdot \mathbf{p}) y' + (t^s_y \cdot \mathbf{r^s_1} \cdot \mathbf{p} - t^s_x \cdot \mathbf{r^s_2} \cdot \mathbf{p}) \quad = \quad 0 \tag{4.29}$$

Esta ecuación describe el lugar de los puntos de la imagen derecha, que pueden corresponder a un punto p de la imagen izquierda. Se puede ver que que el lugar geométrico es la ecuación de

una recta de la forma:

$$a'x' + b'y' + c' \quad = \quad 0 \tag{4.30}$$

Esta recta es llamada línea epipolar derecha. Para cada punto de la imagen izquierda existe una línea epipolar derecha y recíprocamente, para cada punto de la imagen derecha existe una línea epipolar izquierda.

La ecuación 4.29 no es útil puesto que se trata de la ecuación de la línea epipolar en coordenadas de la cámara. En la práctica, se dispone de las coordenadas 2D (u, v) en píxeles (coordenadas en el plano imagen discretizado). Por esta razón, se presenta a continuación la *matriz fundamental* que permite obtener la relación epipolar en coordenadas pixélicas.

4.2.6. La Matriz Esencial

Desarrollando la ecuación 4.29, podemos escribir los parámetros de la recta como:

$$\begin{array}{rcl} a' & = & (t_y^s \cdot r_{31}^s - t_z^s \cdot r_{21}^s)x + (t_y^s \cdot r_{32}^s - t_z^s \cdot r_{22}^s)y + (t_y^s \cdot r_{33}^s - t_z^s \cdot r_{23}^s) \\ b' & = & (t_z^s \cdot r_{11}^s - t_x^s \cdot r_{31}^s)x + (t_z^s \cdot r_{12}^s - t_x^s \cdot r_{32}^s)y + (t_z^s \cdot r_{13}^s - t_x^s \cdot r_{33}^s) \\ c' & = & (t_x^s \cdot r_{21}^s - t_y^s \cdot r_{11}^s)x + (t_x^s \cdot r_{22}^s - t_y^s \cdot r_{12}^s)y + (t_x^s \cdot r_{23}^s - t_y^s \cdot r_{13}^s) \end{array} \tag{4.31}$$

Y puede escribirse en forma matricial como sigue:

$$\begin{pmatrix} a' \\ b' \\ c' \end{pmatrix} = \begin{pmatrix} 0 & -t_z^s & t_y^s \\ t_z^s & 0 & -t_x^s \\ -t_y^s & t_x^s & 0 \end{pmatrix} \begin{pmatrix} r_{11}^s & r_{12}^s & r_{13}^s \\ r_{21}^s & r_{22}^s & r_{23}^s \\ r_{31}^s & r_{32}^s & r_{33}^s \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{4.32}$$

En esta ecuación, el producto de las dos matrices, una antisimétrica (de rango 2) y una matriz ortonormal (de rango 3), generan como resultado la matriz $\mathbf{E}$, llamada *matriz esencial*.

$$\begin{pmatrix} a' \\ b' \\ c' \end{pmatrix} = \quad \mathbf{E} \quad \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{4.33}$$

Esta matriz puede ser calculada a partir de los parámetros t_x^s, t_y^s, t_z^s y r_1^s, r_2^s, r_3^s. Estos parámetros son los que se obtienen en la ecuación 4.17 a partir de los parámetros extrínsecos de cada una de las cámaras (obtenidos tras un proceso de calibración). Esta es la transformación epipolar en la cual a un punto de la imagen izquierda $(x, y, 1)$ le corresponde una recta sobre la imagen derecha descrita por los parámetros (a', b', c').

La ecuación de la recta epipolar se puede escribir en forma matricial de la siguiente manera:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \begin{pmatrix} a' \\ b' \\ c' \end{pmatrix} = 0. \tag{4.34}$$

Sustituyendo en la ecuación 4.33 tenemos:

$$(x, y, 1) \cdot \mathbf{E} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0, \tag{4.35}$$

siendo

$$\mathbf{p}'^t \cdot \mathbf{E} \cdot \mathbf{p} \quad = \quad 0 \tag{4.36}$$

La matriz **E** describe la transformación epipolar izquierda-derecha, la cual permite calcular la ecuación de una línea epipolar que pasa por la imagen derecha asociada a un punto de la imagen izquierda. Observemos que la transformación epipolar derecha-izquierda está dada por la matriz traspuesta:

$$\mathbf{p}^t \cdot \mathbf{E}^t \cdot \mathbf{p}' = 0 \tag{4.37}$$

4.2.7. La Matriz Fundamental

En la práctica, se dispone de la información del entorno en coordenadas de la imagen, por lo que resulta necesario calcular la misma relación izquierda-derecha que describe la matriz esencial, pero con respecto a coordenadas de la imagen.

Ahora vamos a establecer una relación entre un punto de la imagen izquierda y un punto de la imagen derecha (en coordenadas pixélicas). Recordando la relación entre un punto en coordenadas de la cámara y un punto en la imagen en píxeles tenemos:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} f/dx & 0 & u_0 \\ 0 & f/dy & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} sx \\ sy \\ s \end{pmatrix} \tag{4.38}$$

Llamando

$$\mathbf{C} = \begin{pmatrix} f/dx & 0 & u_0 \\ 0 & f/dy & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.39}$$

tenemos

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{C} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{4.40}$$

De la misma forma, para la imagen derecha:

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \mathbf{C}' \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \tag{4.41}$$

Escribiendo la ecuación de forma matricial para la imagen izquierda y llamando $\mathbf{m} = (u, v, 1)^t$ a las coordenadas del punto en la imagen izquierda (en píxeles) y $\mathbf{p} = (x, y, 1)^t$ al mismo punto en coordenadas métricas de la cámara izquierda se obtiene:

$$\mathbf{m} = \mathbf{C} \cdot \mathbf{p} \tag{4.42}$$

y para la imagen derecha:

$$\mathbf{m}' = \mathbf{C}' \cdot \mathbf{p}' \tag{4.43}$$

Sustituyendo en la ecuación 4.36 tenemos:

$$\mathbf{m'^t} \cdot (\mathbf{C^{-1}})^\mathbf{t} \cdot \mathbf{E} \cdot \mathbf{C^{-1}} \cdot \mathbf{m} = 0 \tag{4.44}$$

De la ecuación anterior obtenemos la *matriz fundamental*:

$$\mathbf{F} \quad = \quad (\mathbf{C'^{-1}})^{\mathbf{t}} \cdot \mathbf{E} \cdot \mathbf{C^{-1}} \tag{4.45}$$

La ecuación:

$$\mathbf{m'^{t}} \cdot \mathbf{F} \cdot \mathbf{m} \quad = \quad 0 \tag{4.46}$$

es la ecuación de una línea epipolar en el sistema de coordenadas de la imagen y no en el sistema de coordenadas de la cámara como en la ecuación 4.36. Esta ecuación permite calcular la recta epipolar sobre la que realizaremos la búsqueda de la pareja de cada punto de interés de la imagen izquierda.

Por su parte, al igual que la matriz esencial, la matriz fundamental puede carcularse para un sistema estereoscópico en el que cada una de sus cámaras hayan sido previamente calibradas.

4.2.8. Matching o Emparejamiento

Hasta ahora, se ha mostrado cómo es posible recuperar la información tridimensional a partir de imágenes estáticas capturadas mediante dos cámaras calibradas adecuadamente (reconstrucción tridimensional). Pero para ello necesitamos saber qué puntos de la imagen izquierda representan el mismo objeto en la imagen derecha. Como esto no se conoce *a priori*, se debe buscar la pareja de cada punto de interés de la imagen izquierda en la imagen derecha. Esa búsqueda se restringe a una única recta sobre la imagen derecha, haciendo uso de la geometría epipolar. Eso sí, se hace necesario utilizar técnicas que estimen cual va a ser el correspondiente de entre todos los puntos de dicha recta epipolar. A estas técnicas se les conoce como *técnicas de emparejamiento o "matching"* [21].

El emparejamiento de puntos entre dos imágenes se basa en el principio siguiente: un punto de la primera imagen (cámara izquierda p.e.) representa el mismo punto físico que otro punto de la segunda imagen (decimos que los dos puntos se corresponden) si los dos puntos se asemejan [22]. Esta semejanza debe tener en cuenta los puntos vecinos debido al ruido en las imágenes, los ocultamientos, el cambio de punto de vista de las imágenes, etc.

En ausencia de todo conocimiento a priori, la semejanza entre dos porciones de dos imágenes de una misma escena se puede cuantificar gracias a la medición de la *correlación*, la cual es una medida de semejanza. La correlación se calcula entre dos ventanas, generalmente cuadradas, y centradas en el punto que se está poniendo en correspondencia o apareando.

La correlación de dos funciones continuas complejas $f(x)$ y $g(x)$, representada por $f(x) \circ g(x)$ se define como:

$$f(x) \circ g(x) \quad = \quad \int_{-\infty}^{+\infty} f^*(\alpha) g(x+\alpha) d\alpha \tag{4.47}$$

El equivalente discreto a esta ecuación:

$$f[n] \circ g[n] \quad = \quad \sum_{i=0}^{M-1} f^*[i] g[n+i] \tag{4.48}$$

donde $n = 0, 1, 2, \ldots, M-1$ y M es el tamaño del intervalo en el que es evaluada la función.

Para el caso bidimensional se tienen expresiones similares. Así para la correlación de funciones continuas se tiene:

$$f(x,y) \circ g(x,y) \quad = \quad \iint_{-\infty}^{+\infty} f^*(\alpha,\beta) g(x+\alpha, y+\beta) d\alpha d\beta \tag{4.49}$$

y para el caso discreto:

$$f[n,m] \circ g[n,m] = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} f^*[i,j] g[n+i, m+j] \tag{4.50}$$

Existen muchas variantes distintas de la correlación, pero después de diversos estudios se ha encontrado que la que mejores resultados produce es la ZMNCC (Zero Mean Normalized CrossCorrelation) ya que muestra gran robustez e independencia a variaciones en la iluminación [22].

La correlación ZMNCC entre el punto $\mathbf{p}(u,v)$ de una imagen y el punto $\mathbf{p}'(u',v')$ de otra imagen está dada por la fórmula:

$$\text{ZMNCC}(\mathbf{p},\mathbf{p}') = \frac{\sum_{i=-n}^{n} \sum_{j=-n}^{n} A \cdot B}{\sqrt{\sum_{i=-n}^{n} \sum_{j=-n}^{n} A^2 \cdot \sum_{i=-n}^{n} \sum_{j=-n}^{n} B^2}} \tag{4.51}$$

donde A y B se definen como:

$$\begin{aligned} A &= I(u+i, v+j) - \overline{I(u,v)}, \\ B &= I(u'+i, v'+j) - \overline{I(u',v')}, \end{aligned} \tag{4.52}$$

siendo $I(u,v)$ es la intensidad o nivel de gris en el píxel con coordenadas (u,v), e $\overline{I(u,v)}$ es la media de las intensidades de los puntos que se encuentran en la ventana de tamaño $(2n+1) \times (2n+1)$ con centro en (u,v). El emparejamiento busca, para un punto dado en la primera imagen, el punto en la segunda que tenga la mayor respuesta de correlación (maximice ZMNCC$(\mathbf{p},\mathbf{p}')$).

Con esta este método de emparejamiento y la ayuda de la geometría epipolar, se obtienen pares de puntos estéreo correspondientes al mismo punto de la escena 3D. Conociendo las coordenadas en las imágenes izquierda y derecha -. (u,v) y (u',v') respectivamente.- se obtienen las coordenadas 3D resolviendo el sistema de ecuaciones dado por la ecuación 4.22.

En la siguiente figura se muestra un ejemplo de correspondencia estéreo. En la imagen izquierda se observan dos parches característicos de la escena (en color rojo) y en la derecha, la línea verde representa la recta epipolar y los parches emparejados se indican mediante el color azul.

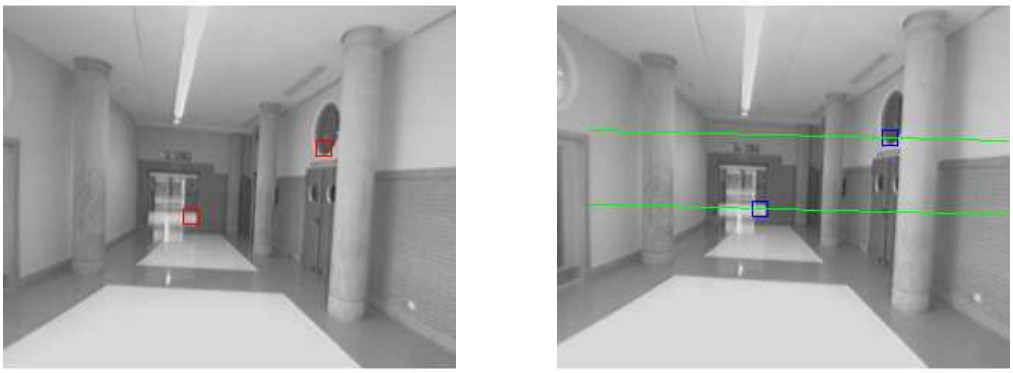


Figura 4.7: Ejemplo de correspondencias estéreo

4.3. Modelo de perspectiva con distorsión

Hasta el momento, solamente se ha utilizado el modelo de cámara ideal pin-hole. Sin embargo en la práctica las lentes introducen deformaciones conocidas como distorsión óptica. Esta es producida cuando los rayos de luz que pasan a través de la óptica son desviados, modificando sus direcciones e interceptando el plano imagen, en posiciones distintas a las que aparecerían en el modelo ideal. Esta desviación aumenta en función de la distancia de cada punto al centro del eje óptico. La distorsión de la lente deteriora la calidad geométrica de la imagen y por tanto la capacidad para medir posiciones de los objetos en la imagen. La distorsión de la lente se puede clasifcar como **radial** y **tangencial**. La distorsión radial de la lente provoca que los puntos de las imágenes se desplacen de forma radial a partir del eje óptico. Su principal causa es un pulido defectuoso de la lente. La distorsión tangencial ocurre en ángulos rectos a las líneas radiales a partir del eje óptico. Su principal causa es un centrado defectuoso de todos los elementos que componen el sistema de lentes [23]. De manera general, los efectos de la distorsión tangencial son menos importantes que los efectos de la distorsión radial. Un ejemplo de ambas distorsiones se puede ver en la figura 4.8 [24] en donde la componentes radial se denota como δ_r y la tangencial comoδ_ϕ

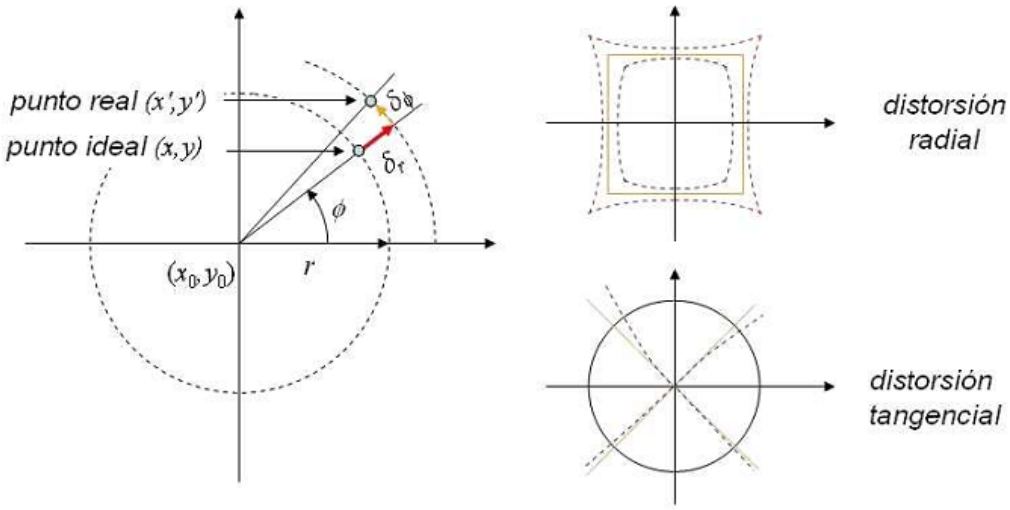


Figura 4.8: Efecto de las componentes de distorsión introducidas por la óptica

La transformación de coordenadas pixélicas a métricas se obtiene a través de la siguiente ecuación:

$$\left\{ \begin{array}{rcl} u & = & x/dx + u_0 \\ v & = & y/dy + v_0 \end{array} \right. \Longrightarrow \left\{ \begin{array}{rcl} x & = & (u - u_0) \cdot dx \\ y & = & (v - v_0) \cdot dy \end{array} \right. \tag{4.53}$$

Si en la ecuación anterior se añaden los términos correspondientes a los errores pixélicos y la distorsión, obtenemos:

$$\left\{ \begin{array}{rcl} x & = & (u + \epsilon_u - u_0) \cdot dx - d0_x \\ y & = & (v + \epsilon_v - v_0) \cdot dy - d0_y \end{array} \right. \tag{4.54}$$

en donde son los errores de medida sobre las coordenadas pixélicas u y v respectivamente, y $d0_x$, $d0_y$ son las componentes de distorsión óptica en x e y, que se pueden expresar en sus

componentes radial y tangencial:

$$\begin{cases} d0_x &= d0_{xr} + d0_{xt} \\ d0_y &= d0_{yr} + d0_{yt} \end{cases} \tag{4.55}$$

Los efectos de la distorsión radial y tangencial se pueden modelar mediante polinomios de orden par a partir de las siguientes expresiones [25]:

$$\begin{cases} d0_{xr} &= (u - u_0) \cdot d_x \cdot (a_1 r^2 + a_2 r^4 + a_3 r^6) \\ d0_{yr} &= (v - v_0) \cdot d_y \cdot (a_1 r^2 + a_2 r^4 + a_3 r^6) \end{cases} \tag{4.56}$$

$$\begin{cases} d0_{xt} &= p_1 \left[r^2 + 2(u - u_0)^2 \cdot dx^2\right] + 2 \cdot p_2 (u - u_0) \cdot d_x \cdot (v - v0) \cdot d_y \\ d0_{yt} &= p_2 \left[r^2 + 2(v - v_0)^2 \cdot dy^2\right] + 2 \cdot p_1 (u - u_0) \cdot d_x \cdot (v - v0) \cdot d_y \end{cases} \tag{4.57}$$

donde a_1, a_2, a_3 son los coeficientes del polinomio que modela la distorsión radial, p_1, p_2 son los coeficientes del polinomio que modela la distorsión tangencial, y el parámetro r es la distancia del punto con coordenadas (u, v) al punto central de la imagen con coordenadas (u_0, v_0), es decir:

$$r = \sqrt{(u - u_0)^2 \cdot dx^2 + (v - v_0)^2 \cdot dy^2} \tag{4.58}$$

A partir de las ecuaciones 4.12, 4.54, 4.55, 4.56 y sustituyendo en la ecuación 4.9, se obtiene la formulación general del modelo de perspectiva con distorsión:

$$\begin{cases} u + \epsilon_u = u_0 + d0_{xr} + d0_{xt} + f_x \cdot \frac{r_{11}X + r_{12}Y + r_{13}Z + T_x}{r_{31}X + r_{32}Y + r_{33}Z + T_z} = P(\Phi) \\ v + \epsilon_v = v_0 + (d0_{yr} + d0_{yt}) \cdot \frac{f_x}{f_y} + f_y \cdot \frac{r_{21}X + r_{22}Y + r_{23}Z + T_y}{r_{31}X + r_{32}Y + r_{33}Z + T_z} = Q(\Phi) \end{cases} \tag{4.59}$$

Φ es un vector de 15 parámetros, 9 de ellos intrínsecos, y 6 extrínsecos:

$$\Phi = [u_0, v_0, a_1, a_2, a_3, p_1, p_2, f_x, f_y, \alpha, \beta, \gamma, T_x, T_y, T_z]^t \tag{4.60}$$

La corrección de la distorsión implica necesariamente una pérdida de precisión métrica, debido al proceso de interpolación. Pero sin embargo, esta corrección es imprescindible desde el punto de vista de búsqueda de correspondencias entre un par estéreo de imágenes. La corrección implica dos pasos: rectificación de las coordenadas pixélicas e interpolación de los niveles de gris de cada píxel. Para acelerar el proceso de corrección de distorsión, la rectificación de la posición de los píxeles debida a la distorsión se calcula de forma previa al inicio, almacenándose en memoria en una *look-up-table* (LUT). Esto es posible gracias a que se dispone de una estructura estéreo calibrada, en la que se conocen tanto los parámetros intrínsecos (en donde se incluyen los coeficientes de distorsión) como los parámetros extrínsecos a partir de las ecuaciones 4.56, 4.57. Por lo tanto, el proceso de corrección de la distorsión solo implica la asignación directa de la nueva posición de cada píxel mediante indexación en la LUT y la ejecución de una serie de interpolaciones de valores en niveles de gris. El mencionado proceso de calibración se realiza previamente de manera off-line, como se comentará posteriormente en la sección 4.4.

Una vez que se han corregido los efectos de la distorsión radial y tangencial, se puede pasar a buscar correspondencias de puntos entre las dos imágenes como se indica en 4.2.3. Esta metodología de eliminar los efectos de la distorsión utilizando LUT es utilizada en [24].

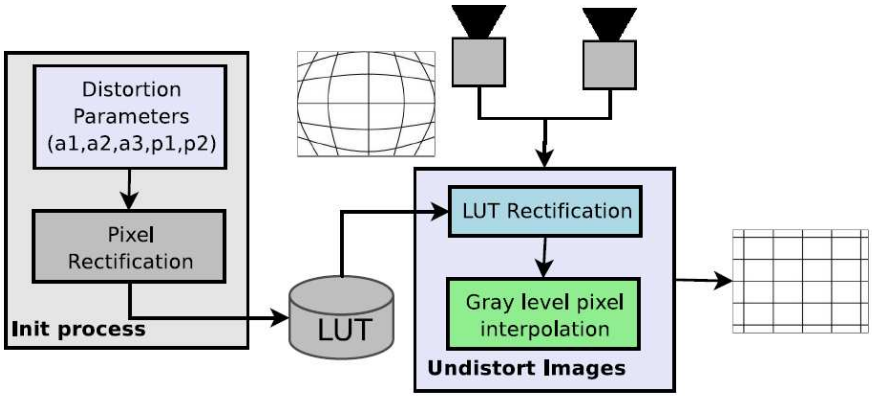


Figura 4.9: Esquema general del proceso de corrección de la distorsión óptica en las imágenes

En otros trabajos estudiados como [11], [4], la manera de proceder es distinta. En estos trabajos no se rectifican las imágenes distorsionadas, sino que en el momento de obtener la posición 3D de una marca característica se procede de la siguiente manera: Para obtener la posición 3D primeramente, se buscan las correspondencias de puntos característicos entre las dos imágenes distorsionadas a lo largo de la línea epipolar en la imagen derecha. Posteriormente, para las coordenadas pixélicas obtenidas en la correspondencia anterior (considerando las imágenes distorsionadas), se les aplica una corrección utilizando un algoritmo iterativo, que permita obtener las coordenadas equivalentes sin los efectos de la distorsión. Una vez que se han corregido las coordenadas, se pasa a obtener la posición 3D de la marca de manera similar a como se indica en la sección 4.2.3.

Estrictamente hablando, **la geometría epipolar es aplicable solamente si no se tienen en cuenta los efectos de la distorsión**. Ya que si se intenta aplicar una búsqueda de correspondencias entre ambas imágenes distorsionadas, la recta epipolar no sería una recta, sino que debería ser una curva debido a la distorsión radial y tangencial. Para solventar este problema lo que se hace es permitir una mayor zona de búsqueda en vertical a lo largo de la recta epipolar, para posteriormente aplicar el método de corrección de distorsión. Aunque en un futuro se desea realizar un estudio en mayor profundidad y una comparativa estricta de ambos métodos, es importante resaltar que ambos métodos convergen, obteniendo buenas correspondencias estéreo, si bien, desde un punto de vista teórico es más razonable el método utilizado en [24] y en el presente trabajo. Las principales diferencias entre un método y otro pueden residir en tiempo de cómputo, aunque esta diferencia de tiempos debería ser muy pequeña.

4.4. Calibración del sistema de visión

La calibración de un sistema de visión [18] consiste en determinar la relación matemática existente entre las coordenadas tridimensionales (3D) de los puntos de una escena y las coordenadas bidimensionales (2D) de esos mismos puntos proyectados y detectados en una imagen. En el modelo pin-hole (apartado 4.1) esta relación viene dada por la siguiente ecuación:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \tag{4.61}$$

La determinación de esta relación es una etapa imprescindible en Visión, en particular para la reconstrucción tridimensional, ya que permite deducir la información tridimensional a partir de las características extraídas de la imagen. En nuestro sistema de odometría visual también es necesario realizar una buena calibración, puesto que se realiza un "tracking" o seguimiento de puntos 3D a partir de las imágenes extraídas por el par estéreo de cámaras.

El ánalisis completo de la calibración de un sistema de visión comprende un estudio de los fenómenos fotométrico, óptico y electrónico, presentes en la cadena de adquisición de imágenes.

En general, un sistema de calibración de una cámara está compuesto por:

1. Un patrón de calibración constituído por unos puntos de referencia conocidos (un tablero de ajedrez por ejemplo).
2. Un sistema de adquisición de imágenes que numere y memorice las imágenes del patrón de calibración.
3. Un algoritmo que establezca la correspondencia entre los puntos 2D detectados en la imagen con los homólogos en el patrón.
4. Un algorimo que calcule la matriz **M** de transformación de perspectiva de la cámara con la referencia asociada al patrón de calibración contenido en la imagen.

Debemos señalar que el proceso de calibración en general, es un proceso supervisado que se ejecuta *off-line*. Una vez calibrada la cámara, se utilizan los parámetros obtenidos de forma *on-line*. Si se produce variación alguna, por ejemplo desde la distancia focal hasta la apertura del objetivo, la cámara tiene que volver a ser calibrada.

En esta sección se mostrarán los resultados obtenidos al calibrar el par estéreo de cámaras utilizado en este trabajo. Dicha calibración se realizó a partir de un conjunto de imágenes (calibración multi-imagen) que mostraban un patrón de calibración en forma de tablero de ajedrez. Para ello se utilizó la *Camera Calibration Toolbox de Matlab* [26].

4.4.1. Extracción de esquinas

La primera fase de la calibración con la Toolbox de Matlab, después de cargar las imágenes que se van a utilizar, consiste en extraer las esquinas del patrón de calibración que en este caso se trata de un patrón tipo *chessboard* o tablero de ajedrez.

Se elige el tamaño de la ventana de búsqueda de las esquinas (por defecto 7×7) y después se realiza un proceso de fijación de las esquinas, para todas las imágenes. Para ello, es necesaria

nuestra participación, ya que la Toolbox de Calibración requiere que se le fijen las esquinas externas del patrón de calibración, así como la distancia de cada lado de un cuadrado del patrón (en nuestro caso 100mm). En la figure 4.10 se puede observar el resultado de la extracción de esquinas para una de las imagenes utilizadas.

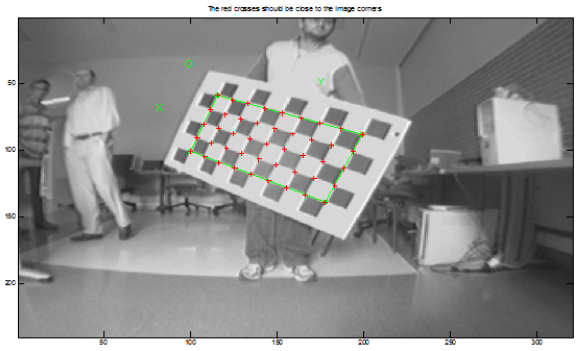


Figura 4.10: Extracción Final de Esquinas para una Imagen

4.4.2. Calibración y resultados

Una vez extraídos todos los puntos del patrón de calibración "chessboard" de todas las imágenes se procede a calibrar la cámara. La calibración se realiza en dos pasos. El primero consiste en una inicialización del vector de parámetros que halla una solución basada en una aproximación lineal, es decir, no tiene en cuenta los parámetros de distorsión. El segundo paso, y a partir de la inicialización anterior, consiste en una optimización no lineal que minimiza la proyección del error según el método de mínimos cuadrados. Las imágenes utilizadas durante la calibración, tienen una resolución de 320 x 240 píxels y son imágenes en escala de grises.

Los parámetros intrínsecos que necesitamos obtener son los siguientes:

- fc: distancia focal en x e y, es decir, son los valores de f/dx y f/dy.
- cc: coordenadas del punto principal o central de la imagen, es decir, u_0 y v_0.
- $[a_1 \;\; a_2 \;\; a_3]$: vector de parámetros de distorsión radial.
- $[p_1 \;\; p_2]$: vector de parámetros de distorsión tangencial.

Una vez realizada la calibración estándar para ambas cámaras de forma independiente, se procede a realización de la calibración del par estéreo. Este método utiliza los parámetros obtenidos en las calibraciones independientes para obtener los parámetros extrínsecos del par estéreo. A su vez, mediante un procedimiento de optimización se recalculan los parámetros de ambas cámaras, minimizando así el error de reproyección.

En el caso de este trabajo, se obtuvieron los siguientes parámetros extrínsecos estéreo:

$$\mathbf{M^s_{ext}} = \begin{pmatrix} r^s_{11} & r^s_{12} & r^s_{13} & t^s_x \\ r^s_{21} & r^s_{22} & r^s_{23} & t^s_y \\ r^s_{31} & r^s_{32} & r^s_{33} & t^s_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0{,}9986 & 0{,}0132 & 0{,}0507 & -153{,}54069 \\ -0{,}0131 & 0{,}9999 & -0{,}0028 & -1{,}50832 \\ -0{,}0507 & 0{,}0021 & 0{,}9987 & -2{,}23524 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Los parámetros intrínsecos recalculados para la cámara izquierda son:

$$\mathbf{M}^{\mathbf{l}}_{\mathbf{int}} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 200{,}0025 & 0 & 160{,}37501 \\ 0 & 202{,}26428 & 129{,}45193 \\ 0 & 0 & 1 \end{pmatrix}$$

- Distorsión radial: $[a_1 \;\; a_2 \;\; a_3] = [-0{,}32997 \;\; 0{,}11324 \;\; 0{,}0]$
- Distorsión tangencial: $[p_1 \;\; p_2] = [-0{,}00047 \;\; -0{,}00067]$

Y respectivamente para la cámara derecha:

$$\mathbf{M}^{\mathbf{r}}_{\mathbf{int}} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 201{,}16081 & 0 & 161{,}03108 \\ 0 & 203{,}2612 & 126{,}52350 \\ 0 & 0 & 1 \end{pmatrix}$$

- Distorsión radial: $[a_1 \;\; a_2 \;\; a_3] = [-0{,}32513 \;\; 0{,}10625 \;\; 0{,}0]$
- Distorsión tangencial: $[p_1 \;\; p_2] = [0{,}00023 \;\; -0{,}00081]$

Una vez obtenidos los parámetros intrínsecos de las cámaras y los parámetros extrínsecos estéreo, disponemos de las herramientas necesarias para obtener información tridimensional a través de las imágenes adquiridas. Podemos visualizar la configuración espacial de los planos de calibración con respecto al par estéreo, como se puede observar en la figura 4.11.

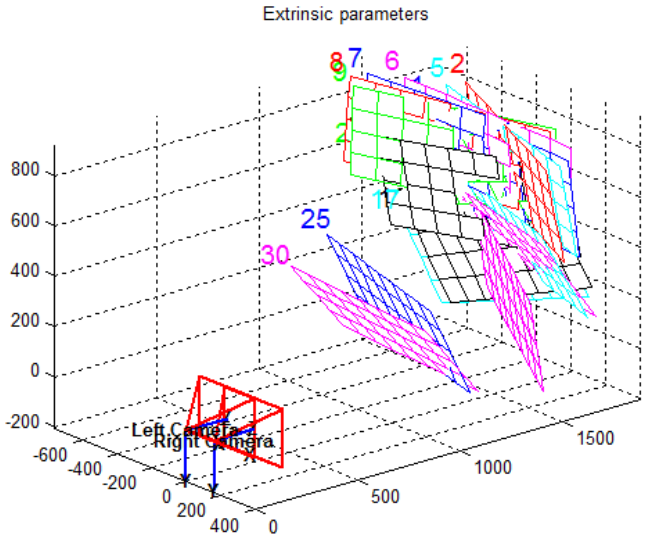


Figura 4.11: Configuración espacial de los planos de calibración con respecto al sistema estéreo

Capítulo 5

Extracción de Marcas Naturales

El mapa del entorno generado a partir de las técnicas de SLAM, está formado por la posición 3D de una serie de **marcas naturales** consideradas como **estáticas**. En este escenario, la extracción de características en una imagen es el proceso que identifica estas marcas naturales en el entorno 3D. Lo que se desea de estas marcas, es que sean fácilmente detectables a lo largo del tiempo, con el fin de evitar añadir al sistema un mayor número de marcas, ya que el tiempo de cómputo en el EKF-SLAM aumenta en función del número de marcas existentes en el mapa.

Para una imagen determinada, surgen las siguientes preguntas: **¿Cómo escoger los puntos característicos de una imagen?**, **¿cómo se debe realizar el tracking de estos puntos a lo largo del tiempo?**. Una extracción pobre de características puede conducir a obtener pocos y malos puntos de interés, por lo que luego si estos puntos se utilizan como marcas naturales, esto puede ocasionar resultados catastróficos ya que el algoritmo de SLAM puede llegar a divergir. Por lo tanto, es crucial realizar un estudio previo de diversos algoritmos de extracción de características, para ver el rendimiento de los distintos algoritmos estudiados.

Por otra parte, un *punto de interés* es un punto en la imagen en el que existe una variación alta de intensidad tanto en horizontal como en vertical. De manera general, las características que presenta un punto de interés en la imagen son:

- Tiene una posición en la imagen bien definida.
- La estructura local alrededor del punto de interés es rica en términos de *contenido de información* local. Esto hace que el uso de los puntos de interés simplifique el procesado en un sistema basado en visión.
- Es *estable* bajo perturbaciones en la imagen tanto globales como locales. Entre éstas se incluyen deformaciones procedentes tanto de transformaciones de perspectiva (transformaciones afines, cambios de escala, rotaciones y/o traslaciones, ...) como de variaciones en la iluminación/brillo. Se dice que los puntos de interés deben poseer un alto grado de *reproducibilidad*.

Generalmente, la noción de punto de interés va unida a la detección de esquinas, donde las esquinas son detectadas con el objetivo principal de obtener características robustas, bien definidas y estables para realizar un seguimiento de objetos y reconocimiento tridimensional de objetos a partir de imágenes bidimensionales.

En este trabajo se han analizado cuatro métodos diferentes de extracción de características de bajo nivel: Shi-Tomasi, esquinas de Harris, un detector de esquinas afín invariante y la diferencia

de Gaussianas (DOG). No se han analizado descriptores de más alto nivel tipo SIFT [27] o SURF [28] cuyo tiempo de cómputo es mucho mayor, ya que estos descriptores serán incorporados en un nivel de SLAM superior en un futuro. De manera general, no se recomienda utilizar estos descriptores invariantes de alto nivel tipo SIFT para un tracking continuo, debido al alto coste computacional que presenta, sin embargo, el uso de estos descriptores es adecuado realizarlo de vez en cuando para mejorar la predicción en la posición del móvil [4].

A continuación se comentarán las características principales de cada uno de ellos. Los resultados de la comparativa se muestran en la sección 7. Cabe mencionar, que el detector de marcas naturales no es empleado de una manera continua, sino que solamente se utiliza cuando es necesario añadir más marcas al sistema, ya que luego se realiza un seguimiento o tracking de las marcas añadidas tomando como referencia el parche inicial. En la figura 5.1 se muestra un ejemplo de la extracción de características (esquinas en color verde) en una imagen.

Figura 5.1: Ejemplo de extracción de características en una imagen

5.1. Detección de esquinas basado en la autocorrelación local

Los dos detectores de esquinas más utilizados son el detector de esquinas de Harris [29] y el detector de esquinas de Shi-Tomasi [15]. Ambos detectores se basan en la función de autocorrelación local, que mide los cambios locales de la señal con desplazamientos pequeños en diferentes direcciones.

Dado un desplazamiento $(\triangle u, \triangle v)$ y un punto (u, v) la función de autocorrelación se define como:

$$c(u,v) = \sum_W [I(u_i, v_i) - I(u_i + \triangle u, v_i + \triangle v)]^2 \tag{5.1}$$

donde $I(u, v)$ es la intensidad (nivel de gris) de la imagen en el punto (u, v) y los puntos (u_i, v_i) son los puntos de la ventana W centrada en (u, v).

La imagen desplazada se aproxima por un desarrollo en serie de Taylor truncado a los términos de primer orden:

$$I(u_i + \triangle u, v_i + \triangle v) \approx I(u_i, v_i) + [I_u(u_i, v_i) I_v(u_i, v_i)] \begin{bmatrix} \triangle u \\ \triangle v \end{bmatrix} \tag{5.2}$$

donde I_u e I_v denotan las derivadas parciales con respecto a u e v, respectivamente.

Sustituyendo la ecuación (5.2) en (5.1) tenemos:

$$\begin{aligned}
c(u,v) &= \sum_W [I(u_i, v_i) - I(u_i + \triangle u, v_i + \triangle v)]^2 \\
&= \sum_W \left(I(u_i, v_i) - I(u_i, v_i) - [I_u(u_i, v_i) I_v(u_i, v_i)] \begin{bmatrix} \triangle u \\ \triangle v \end{bmatrix} \right)^2 \\
&= \sum_W \left(-[I_u(u_i, v_i) I_v(u_i, v_i)] \begin{bmatrix} \triangle u \\ \triangle v \end{bmatrix} \right)^2 \\
&= \sum_W \left([I_u(u_i, v_i) I_v(u_i, v_i)] \begin{bmatrix} \triangle u \\ \triangle v \end{bmatrix} \right)^2 \\
&= [\triangle u, \triangle v] \begin{bmatrix} \sum_W (I_u(u_i, v_i))^2 & \sum_W I_u(u_i, v_i) I_v(u_i, v_i) \\ \sum_W I_u(u_i, v_i) I_v(u_i, v_i) & \sum_W (I_v(u_i, v_i))^2 \end{bmatrix} \begin{bmatrix} \triangle u \\ \triangle v \end{bmatrix} \\
&= [\triangle u, \triangle v] \, \mathbf{C}(u,v) \begin{bmatrix} \triangle u \\ \triangle v \end{bmatrix}
\end{aligned} \tag{5.3}$$

donde la matriz $\mathbf{C}(u, v)$ captura la estructura local en intensidad de la vecindad del punto (u, v). Por ello se le denomina *matriz de estructura local*. Esta matriz se puede diagonalizar mediante una matriz de paso ortogonal:

$$\mathbf{C} = \mathbf{R}^{\mathbf{t}} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \mathbf{R}$$

Siendo λ_1, λ_2 los autovalores de la matriz $\mathbf{C}(u, v)$. Dichos autovalores forman una descripción invariante a la rotación.

Se puede demostrar que ambos autovalores son positivos o cero (la matriz es definida o semidefinida positiva). A partir de estos autovalores se puede extraer la siguiente interpretación geométrica (siendo $\lambda_1 > \lambda_2$):

1. Si tanto λ_1 como λ_2 son pequeños, la función de autocorrelación es plana ($c(u, v)$ cambio poco en cualquier dirección). Por lo tanto, la región de la imagen bajo estudio se caracteriza por una intensidad aproximadamente constante. De hecho, para una imagen uniforme ambos $\lambda_1 = \lambda_2 = 0$.

2. En la localización de un *borde*, $\lambda_1 > 0$, $\lambda_2 \approx 0$. El autovector correspondiente a λ_1 sigue la dirección normal al borde (dirección de gran incremento de la función de autocorrelación).

3. Una esquina produce dos autovalores positivos ($\lambda_1 \geq \lambda_2 > 0$). Conforme más grandes son los autovalores, mayor es el contraste de los bordes ortogonales a las direcciones de los correspondientes autovectores.

5.1.1. Detector de Shi-Tomasi

El método desarrollado se fundamenta en el trabajo original de Shi-Tomasi [15]. Este método se basa en el uso del gradiente de intensidad para cada píxel, el cual proporciona información sobre la *no uniformidad* en los niveles de gris a lo largo de la imagen. Se tienen en cuenta tanto bordes verticales como horizontales, buscando aquellos puntos que presenten cambios bruscos en ambos gradientes.

Para ver el grado de no uniformidad, se propone el uso de la siguiente matriz aplicada al conjunto de píxels del path, siendo I_u el gradiente de intensidad de la imagen en horizontal, y I_v el gradiente en vertical.

$$Z = \sum_{y=v_{0parche}}^{v_{max\ parche}} \sum_{x=u_{0\ parche}}^{u_{max\ parche}} \begin{pmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{pmatrix} \tag{5.4}$$

Las expresiones utilizadas para calcular los gradientes vertical y horizontal son las siguientes:

$$\begin{cases} I_x = \frac{I(x+1,y) - I(x-1,y)}{2} \\ I_y = \frac{I(x,y+1) - I(x,y-1)}{2} \end{cases} \tag{5.5}$$

Es decir, Z implica el cálculo de los gradientes en todos los píxels del parche, la formación de las matrices individuales y su posterior suma.

Para poder evaluar el grado de no uniformidad del parche, es necesario calcular los dos autovalores λ_1 λ_2 de la matriz Z. Cada uno de los autovalores proporciona información sobre la no uniformidad de la imagen en una determinada dirección. Un parche será seleccionado si el menor de los autovalores calculados λ_1 o λ_2 presenta un valor elevado. En los casos en los que solamente uno de los dos autovalores es elevado, el parche no es adecuado para ser seleccionado como una marca natural, ya que el parche solamente presenta interés en una sola dirección (por ejemplo un borde), pero no en la dirección perpendicular a esta. Por lo tanto, solamente serán seleccionados aquellos parches cuyos autovalores sean elevados, lo que significa que el parche presenta interés tanto en horizontal como en vertical (por ejemplo una esquina).

Para calcular la Z total del parche en todos los puntos (x, y) de la región de búsqueda, el procedimiento optimizado es el siguiente:

1. En primer lugar se calculan las sumas de las Z individuales de píxel por columnas a lo ancho de todo el área de búsqueda, tomando como número de filas la altura del parche B (ver figura 5.2).

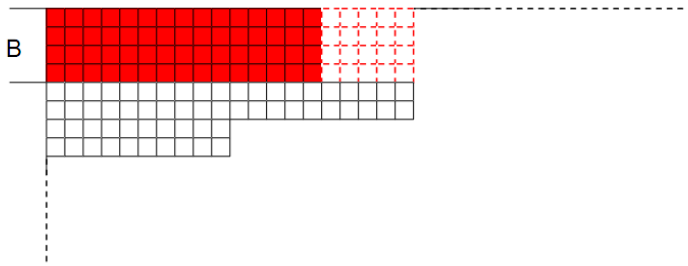


Figura 5.2: Cálculo de Z: Paso 1

2. Posteriormente, se calcula la Z total del primer parche a evaluar sumando los resultados anteriores para las B primeras columnas (ver figura 5.3).

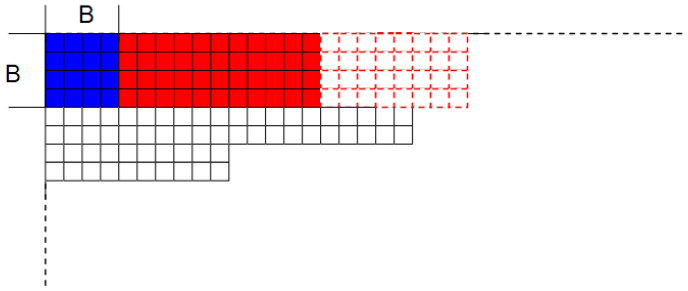


Figura 5.3: Cálculo de Z: Paso 2

3. A partir de este punto se realiza un proceso iterativo en el que para todo valor de $\boldsymbol{u}$ de la zona de búsqueda, tomando como referencia el último valor de Z calculado, se resta el valor de la suma correspondiente a la columna anterior y se suma el valor correspondiente a la columna posterior. De esta forma, queda calculado el valor de Z total para el siguiente parche a evaluar (ver figura 5.4).

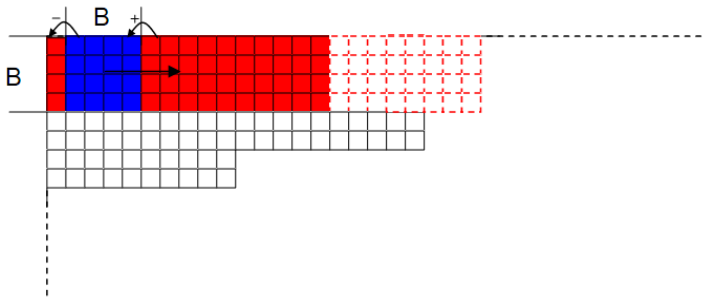


Figura 5.4: Cálculo de Z: Paso 3

5.1.2. Detector de Harris

Para el detector de Harris, la matriz de estructura local se suaviza con un filtro Gaussiano:

$$\mathbf{C}(u,v) = w_G(\sigma) * \begin{bmatrix} \sum_W (I_x(x_i,y_i))^2 & \sum_W I_x(x_i,y_i) I_y(x_i,y_i) \\ \sum_W I_x(x_i,y_i) I_y(x_i,y_i) & \sum_W (I_y(x_i,y_i))^2 \end{bmatrix} \tag{5.6}$$

donde $w_G(\sigma)$ es un filtro Gaussiano con desviación estándar σ y la operación $*$ denota convolución. Se define la siguiente medida que indica lo bueno que es una esquina en función de la matriz de estructura:

$$r(x,y) = |\mathbf{C}(x,y)| - k\ [\text{traza}(\mathbf{C}(x,y))]^2, \tag{5.7}$$

donde k es un parámetro ajustable y $\mathbf{C}(x,y)$ es la matriz de estructura local en las coordenadas (x,y).

Para prevenir que se detecten muchas esquinas demasiado juntas, normalmente se implementa un proceso de *supresión no máxima* que suprime esquinas débiles que rodean a esquinas de mejor calidad. A esto le sigue posteriormente un proceso de umbralizado.

En definitiva, el detector de esquinas de Harris depende de los siguientes parámetros:

- Parámetro k. Se demuestra que $r(x,y) > 0 \rightarrow 0 \leq k \leq 0{,}25$. Cuanto mayor es k la respuesta $r(u,v)$ es menor y por tanto menos esquinas se detectan. Si embargo, si k aumenta el sistema es más inmune al ruido.
- Radio d_{min}: mínima distancia entre esquinas detectadas.
- Umbral t. Si $r(x,y) > t$, el punto (u,v) es esquina.

Posteriormente, se itera un algorimo que obtiene con precisión la posición subpixélica de las esquinas en la imagen. La idea de este algoritmo, se basa en la observación de todo vector dedsde el centro q hasta un punto p localizado en una vecindad de q es orogonal al gradiente de la imagen en el punto p sujeto al ruido de medida y ruido de la imagen.

$$\epsilon_i \ = \ \nabla I_{p_i}^T \cdot (q - p_i) \tag{5.8}$$

donde ∇I_p es el gradiente de la imagen en uno de los puntos p en la vecindad de q. El valor de q es aquel que minimiza ϵ_i. Se puede obtener un sistema de ecuaciones igualando los ϵ_i a 0:

$$\left(\sum_i \nabla I_{pi} \cdot \nabla I_{pi}^T \right) \cdot q - \left(\sum_i \nabla I_{pi} \cdot \nabla I_{pi}^T \cdot p_i \right) = 0 \tag{5.9}$$

En la ecuación 5.9 los gradientes se suman en un entorno de vecindad o ventana de búsqueda de q. Si se denomina el primer término de la ecuación como G y el segundo término como b, se obtiene la ecuación $q = G^{-1} \cdot b$. El algoritmo fija el centro de la ventana de búsqueda en este nuevo centro q y luego itera hasta que el centro se mantiene bajo un cierto umbral.

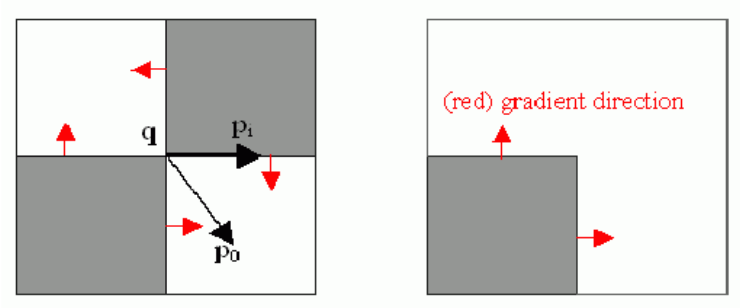


Figura 5.5: Obtención de esquinas con precisión subpixélica

5.2. Detector de Esquinas Afín Invariante

Las esquinas en una imagen son características importantes que pueden ser utilizadas para realizar matching, reconstrucción estéreo y/o flujo óptico. Los detectores anteriormente comentados, no cumplen la propiedad de invarianza frente a transformaciones afines (es decir, las esquinas no pueden ser detectadas desde todos los ángulos posibles). En esta sección se desarrollan los fundamentos de un detector de esquinas afín invariante multiescala, basado en las derivadas de segundo orden de la intensidad de la imagen [30].

Para poder implementar el citado detector invariante, se realiza un cambio de las coordenadas pixélicas (x, y) a un nuevo sistema de coordenadas denominado **coordenadas gauge**. A continuación se describe en que consiste el cambio de coordenadas y otros aspectos necesarios para la formulación del detector.

5.2.1. Coordenadas Gauge

Lo que nos interesa en este momento es obtener un punto en una imagen representado de tal forma, que si se conserva la misma imagen local alrededor de este punto, sin importar la rotación, la descripción del punto siempre sea la misma.

Esto se puede lograr, mediante el cambio de coordenadas pixélicas de un punto en la imagen, a un sistema de coordenadas que dependa de las derivadas direccionales de la imagen local. Se define cada punto de la imagen original en un nuevo sistema de coordenadas locales $(\vec{w}, \vec{v})$. Estas coordenadas se fijan de tal modo que $(\vec{w})$ apunte en la dirección de máximo cambio de intensidad, y $(\vec{v})$ sea perpendicular 90^o a $\vec{w}$. Por lo tanto este nuevo sistema de coordenadas queda definido por la siguiente transformación:

$$\begin{aligned} \vec{w} &= \left(\frac{\partial L}{\partial x}, \frac{\partial L}{\partial y}\right) \\ \vec{v} &= \left(\frac{\partial L}{\partial y}, -\frac{\partial L}{\partial x}\right) \end{aligned} \tag{5.10}$$

En la ecuación 5.10 L significa el valor de luminancia de la imagen para un punto (x, y). En la figura 5.6 se puede observar las coordenadas gauge locales de primer orden. El vector unitario $\vec{v}$ es siempre tangencial a las líneas de intensidad constante, mientras que el vector unitario $\vec{w}$ es siempre perpendicular a las líneas de intensidad constante y a los puntos en la dirección del vector gradiente.

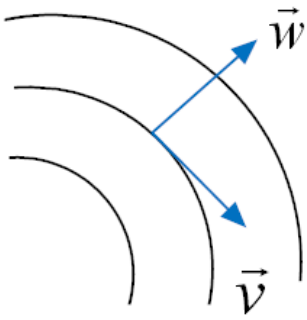


Figura 5.6: Coordenadas gauge de primer orden

Para poder calcular derivadas respecto a las coordenadas gauge, siempre es necesario partir de las coordenadas cartesianas (pixélicas) y posteriormente aplicar el cambio de coordenadas. Para ello es necesario acudir a la definición de derivada direccional en donde $L_x = \frac{\partial L}{\partial x}$, $L_y = \frac{\partial L}{\partial y}$.

$$\vec{w} \cdot \vec{\nabla} = \vec{w} \cdot \{L_x, L_y\} \tag{5.11}$$

Por ejemplo, para obtener las derivadas respecto a las coordenadas gauge de primer orden, se procede de la siguiente manera:

$$\begin{aligned} \frac{\partial L}{\partial \vec{w}} = L_w = \vec{w} \cdot \vec{\nabla} = \vec{w} \cdot \{L_x, L_y\} = \frac{L_x^2 + L_y^2}{\sqrt{L_x^2 + L_y^2}} = \sqrt{L_x^2 + L_y^2} \\ \frac{\partial L}{\partial \vec{v}} = L_v = \vec{v} \cdot \vec{\nabla} = \vec{v} \cdot \{L_x, L_y\} = \frac{L_y L_x - L_x L_y}{\sqrt{L_x^2 + L_y^2}} = 0 \end{aligned} \tag{5.12}$$

Si ahora tomamos derivadas respecto a las coordenadas gauge de primer orden, como están fijadas al objeto, sin importar la rotación o la traslación, se obtienen los siguientes resultados:

- Cualquier derivada expresada en coordenadas gauge es un invariante ortogonal [31]. La derivada de primer orden $\frac{\partial L}{\partial \vec{w}}$ es la derivada en la dirección del gradiente, y por lo tanto se trata del mismo gradiente que es un invariante.
- Como $\frac{\partial L}{\partial \vec{v}} = 0$ implica que no existe ningún cambio en la luminancia a medida que nos movemos tangencialmente sobre las líneas de intensidad constante.

Si se desea consultar más información sobre como obtener derivadas de coordenadas gauge de un orden mayor, así como las interpretaciones de los resultados de las derivadas de mayor orden, se puede consultar la referencia [32].

5.2.2. Curvatura de las Isolíneas

Como se ha comentado anteriormente, las isolíneas son aquellas líneas de la imagen que presentan la misma intensidad. La curvatura de las isolíneas k se define como el cambio $w^{''} = \frac{\partial^2 w}{\partial v^2}$ del vector tangente $w^{'} = \frac{\partial w}{\partial v} = v$ en el sistema de coordenadas gauge.

De la definición de isolínea se tiene que $L(v, w) = Constante$, y $w = w(v)$. A partir de estas consideraciones, si diferenciamos obtenemos la siguiente expresión:

$$\frac{\partial L(v, w(v))}{\partial v} = w^{'}(v) \cdot L_w(v, w(v)) + L_v(v, w(v)) = 0 \tag{5.13}$$

Despejando de la ecuación 5.14 y como sabemos que por definición $L_v = 0$, podemos obtener la expresión de la primera derivada $w^{'}$:

$$w^{'} = -\frac{L_v}{L_w} = 0 \tag{5.14}$$

Si volvemos a diferenciar la ecuación 5.13 obtenemos la expresión para la curvatura de las isolíneas en coordenadas gauge:

$$k = w^{''} = -\frac{L_{vv}}{L_w} \tag{5.15}$$

si expresamos la ecuación 5.15 en coordenadas cartesianas tenemos:

$$k = -\frac{-2 \cdot L_x L_{xy} L_y + L_{xx} L_y^2 + L_x^2 L_{yy}}{\left(L_x^2 + L_y^2\right)^{3/2}} \tag{5.16}$$

5.2.3. Formulación del Detector

Las esquinas en una imagen se definen por zonas en las cuáles las isolíneas presentan una alta curvatura y con un valor alto de grandiente de intensidad. El siguiente detector ha sido propuesto por Blom en [30]. En dicho trabajo, se plantea elevar el producto de la expresión de la curvatura de las isolíneas y el gradiente L_w a una potencia determinada n:

$$\Theta[n] = -\frac{L_{vv}}{L_w} \cdot L_w^n = k \cdot L_w^n = -L_{vv} \cdot L_w^{n-1} \tag{5.17}$$

Una de las principales características de este detector es la invarianza frente a transformaciones afines, lo que permite que una esquina sea detectada desde todos los ángulos posibles. Una transformación afín, es una transformación lineal de los ejes de coordenadas:

$$\begin{pmatrix} x^{'} \\ y^{'} \end{pmatrix} = \frac{1}{ad - bc} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \tag{5.18}$$

Se puede omitir el término $(e \quad f)$ y estudiar solamente la propiedad de transformación afín. El término $\frac{1}{ad-bc}$ es el determinante de la matriz de transformación, y su propósito es ajustar la amplitud cuando el área cambia. Si aplicamos el concepto de transformación afín a las derivadas de primer orden, obtenemos la siguiente expresión:

$$\begin{pmatrix} \partial x^{'} \\ \partial y^{'} \end{pmatrix} = \frac{1}{ad - bc} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \partial x \\ \partial y \end{pmatrix} \tag{5.19}$$

Si aplicamos la transformación afín a la definición de las coordenadas gauge, obtenemos una ecuación para las coordenadas transformadas:

$$-L_{v_a v_a} \cdot L_{w_a}^{n-1} = \frac{\left(\frac{\left(a^2+c^2\right)L_x^2+2(ab+cd)L_xL_y+\left(b^2+d^2\right)L_y^2}{(bc-ad)^2}\right)^{\frac{1}{2}(-3+n)} \left(2 \cdot L_x L_{xy} L_y - L_{xx} L_y^2 - L_x^2 L_{yy}\right)}{(bc-ad)^2} \tag{5.20}$$

En la ecuación anterior, se puede comprobar que cuando $n = 3$ y para una transformación afín para la cual $bc - ad = 1$, la transformación obtenida es independiente de los parámetros a, b, c y d por lo que se cumple la condición de invarianza. Reordenado la ecuación 5.20 y con las consideraciones de $n = 3$ y $bc - ad = 1$, obtenemos la siguiente expresión:

$$\Theta[n] \;=\; -\frac{L_{vv}}{L_w} \cdot L_w^3 \;= L_{vv} L_w^2 = 2 \cdot L_x L_{xy} L_y - L_{xx} L_y^2 - L_x^2 L_{yy} \tag{5.21}$$

La ecuación anterior proporciona la expresión de un detector de esquinas invariante ante transformaciones afines. Además, presenta la propiedad de que es **no singular** en aquellos puntos en los que el gradiente es igual a 0, y debido a su propiedad de invarianza, puede detectar esquinas desde todos los *ángulos posibles*.

Además, el presente operador es un operador multiescala. La imagen original $L(x, y)$ es convolucionada con un kernel Gaussiano 2D en el cuál el parámetro σ (ancho del kernel Gaussiano) representa la escala con la que realizamos la observación. La expresión general para un kernel Gaussiano de 2D es la siguiente:

$$G_{2D}(x, y; \sigma) \;=\; \frac{1}{2\pi\sigma^2} e^{-\frac{\left(x^2+y^2\right)}{2\sigma^2}} \tag{5.22}$$

Sin embargo, en este trabajo no se realiza propiamente un análisis multiescala, sino que **se asume una única escala fija fina**, la cuál es representativa de la longitud de las estructuras presentes en las imágenes. Esto es debido, a que en el nivel métrico de SLAM que nos encontramos, nos interesa un detector de esquinas rápido y que sea capaz de obtener puntos característicos en la imagen.

Para escalas finas, las respuestas más altas del detector se obtienen para esquinas pronunciadas y para un pequeño número de falsas perturbaciones de escala fina a lo largo de los bordes. Luego a medida que se incrementa el parámetro de escala σ la selectividad a la detección de estructuras tipo unión aumenta. En particular, aquellas esquinas difusas y redondeadas, solamente presentan respuestas altas para escalas grandes. A continuación, se muestran los principales aspectos del detector de esquinas afín invariante con relación a la escala:

- Si solamente estamos interesados en esquinas pronunciadas, esto es, esquinas que se pueden aproximar bien por líneas rectas, es suficiente con utilizar una escala fina en la fase de detección. El motivo de utilizar escalas más grandes, es para reducir el número de falsos positivos.
- Si estamos interesados en detectar esquinas redondeadas y esquinas para las cuáles las variaciones de intensidad a través de los bordes de la esquina son lentas (esquinas difusas), es necesario utilizar una escala grande.

Una propiedad conocida de la representación de imágenes en análisis multiescala, es que la amplitud de las derivadas espaciales en general decrece a medida que la escala aumenta. Esto es

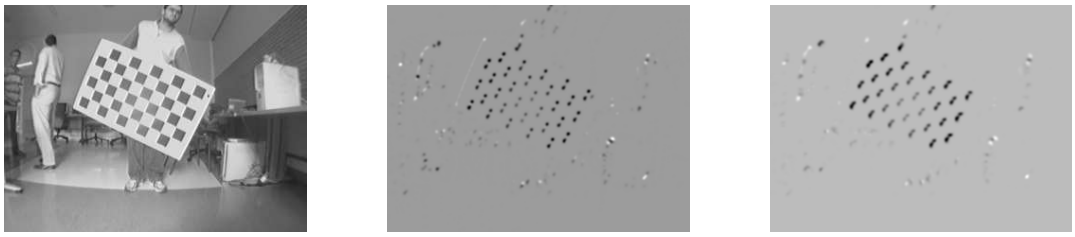
debido a la no invarianza que presentan los operadores diferenciales. Para evitar este problema, se trabaja en las llamadas *coordenadas naturales* [32]. Para ello se considera la transformación $\frac{x}{\sigma} \rightarrow \hat{x}$, luego $\hat{x}$ es adimensional. Al trabajar con coordenadas naturales, se logra la invarianza ante la escala. La formulación de las coordenadas naturales es la siguiente:

$$\frac{\partial^n}{\partial \hat{x}^n} \rightarrow \sigma^n \frac{\partial^n}{\partial x^n} \tag{5.23}$$

Para estudios más profundos en análisis multiescala sobre la selección de la escala óptima, se pueden consultar las referencias [33] [34].

5.2.4. Resultados

En la figura 5.7 se puede observar el resultado de aplicar este operador a una imagen bajo diferentes valores del parámetro de escala σ. La figura 5.7(a) representa la imagen original, mientras que las imágenes 5.7(b) y 5.7(c) representan la detección de esquinas para unos valores de $\sigma = 1$ y $\sigma = 3$ respectivamente. Como se puede ver, se destaca la curvatura de las isolíneas, la curvatura positiva (convexa, en color claro) y la curvatura negativa (cóncava, en color oscuro).



(a) Imagen Original

(b) Detección de esquinas con $\sigma = 1$

(c) Detección de esquinas con $\sigma = 3$

Figura 5.7: Resultados detección de esquinas a diferentes escalas

Sin embargo, a la hora de extraer las esquinas de la imagen, se considera el valor absoluto del detector, sin tener en cuenta la curvatura de las isolíneas. Posteriormente, la imagen es umbralizada con un valor de umbral T y las esquinas son extraídas. A pesar, de que solamente se trabaja con una escala fija fina, el hecho de que la imagen del detector de esquinas sea umbralizada implica el uso de coordenadas naturales para obtener invarianza ante cambios de escala, de modo que si se selecciona otra escala, el valor de umbral T pueda seguir siendo el mismo.

Para determinar las coodenadas del punto correspondiente a una esquina en la imagen de esqunas, se determina el centroide de cada uno de los *blobs* extraídos de la imagen de esquina. Los valores de las coordenadas en la imagen de las esquinas calculadas son:

$$\begin{cases} x_c &= \frac{\sum_{i=1}^{n} L(x_i,yi) \cdot x_i}{\sum_{i=1}^{n} L(x_i,y_i)} \\ \\ y_c &= \frac{\sum_{i=1}^{n} L(x_i,yi) \cdot y_i}{\sum_{i=1}^{n} L(x_i,y_i)} \end{cases} \tag{5.24}$$

En la ecuación 5.24 $L(x_i, y_i)$ es el valor de luminancia de la imagen de esquinas obtenida con el valor absoluto del detector de la ecuación 5.21. Además, se realiza un proceso de *supresión no máxima* con el fin de evitar que se detecten muchas esquinas demasiado juntas. En la figura 5.8 se muestra un ejemplo de la imagen original y de la imagen de esquinas obtenida, que posteriormente será umbralizada y se extraerán las esquinas correspondientes.

(a) Imagen Original (b) Detección de esquinas con $\sigma = 3$

Figura 5.8: Extracción de esquinas para una escala fija

5.3. Diferencia de Gaussianas DOG

En esta sección se expone de manera muy breve los fundamentos del operador de diferencia de Gaussianas (DOG). Se incluye en el presente trabajo para obtener datos comparativos entre los distintos detectores propuestos. Sin embargo, dado a la elevada carga computacional que supone la implementación de este operador, su uso es inviable como detector de características de bajo nivel, tal y como se propone el uso de estos detectores en el presente trabajo.

Este método se basa en la localización de puntos de interés utilizando operadores multiescala, y constituye la primera parte del algoritmo de los descriptores SIFT [27]. El primer paso para detectar puntos de interés es identificar las posiciones y escalas características que pueden ser repetidamente asignadas bajo diferentes puntos de vista del mismo objeto. La detección de puntos que sean invariantes a los cambios de escala de la imagen, se puede conseguir mediante la búsqueda de características estables a través de todas las escalas posibles utilizando una función continua de escala conocida como espacio de escalas o *scale-space*.

Para detectar puntos estables en el espacio de escalas, se utiliza el operador diferencia de Gaussianas convolucionado con la imagen original. Este operador puede ser calculado mediante la diferencia de dos escalas cercanas separadas por un factor constante k:

$$D\left(x, y, \sigma\right) \;=\; \left(G\left(x, y, k\sigma\right) - G\left(x, y, \sigma\right)\right) * I(x, y) \;=\; L\left(x, y, k\sigma\right) - L\left(x, y\right) \tag{5.25}$$

La función diferencia de Gaussianas, proporciona una buena aproximación al operador normalizado Laplaciana de Gaussiana $\sigma^2\nabla^2 G$ [33]. En [34] se demostró que el máximo y el mínimo de $\sigma^2\nabla^2 G$ proporcionan características mas estables comparado con otros posibles detectores como el gradiente, Shi-Tomasi o esquinas de Harris.

En la figura 5.9 se puede observar el proceso de la diferencia de Gaussianas. Para cada octava del espacio de escalas, la imagen inicial es convolucionada con kernels Gaussianos de diferente desviación típica para producir un conjunto de imágenes en el espacio de escalas. Las imágenes de escalas adyacentes son substraídas para producir imágenes de diferencia de Gaussianas.

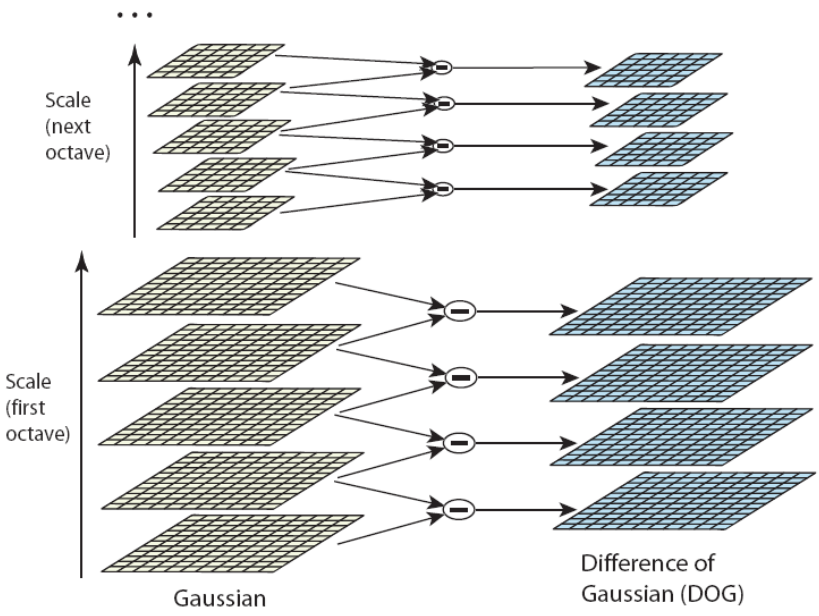


Figura 5.9: Proceso del operador diferencia de Gaussianas

Para detectar los mámimos y mínimos locales de $D(x, y, \sigma)$, cada punto se compara con sus 8 vecinos en la imagen actural y nueve vecinos en la escala superior e inferior. Se selecciona solamente si su respuesta es más grande o más pequeña que la respuesta de todos sus vecinos. Este proceso se puede observar en la figura 5.10.

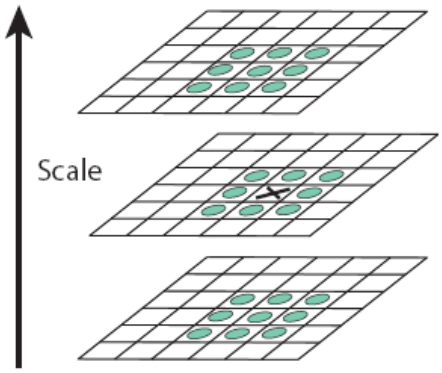


Figura 5.10: Detección de máximos y mínimos en el espacio de escalas

Una vez que se han detectado los máximos y mínimos de este operador a lo largo del espacio de escalas, se seleccionan aquellos puntos que presentan una mayor respuesta y además que esta respuesta supere un determinado umbral. Posterioremente, se aplica un criterio de *supresión no máxima* con el fin de evitar que se detecten muchas marcas demasiado juntas.

Capítulo 6

Visual SLAM

El único sensor utilizado en el presente trabajo, es un sistema de visión estéreo de óptica gran angular movido con la mano, por lo que se trabaja con 6 grados de libertad (6DOF). Se ha realizado un enfoque métrico, en el cuál el estado representa la posición real de la cámara respecto del entorno 3D. Se aplican técnicas de SLAM basadas en visión mediante el uso del EKF. La solución adaptada se basa en los trabajos referenciados en [4] [11].

Uno de los principales problemas a la hora de resolver el problema de SLAM está relacionado con la manera de construir el mapa. Debido a que existen errores acumulativos en las medidas, es posible la aparición de una *deriva* durante el proceso de construcción del mapa. Esta deriva puede llegar a ocasionar que no se reconozcan lugares previamente visitados, degenerando la calidad del mapa 3D del entorno. Por lo tanto, de cara a conseguir una buena localización por largos períodos de tiempo, **se incluye el mapa completo en el proceso del filtro**, es decir, el vector de estado contendrá información de todas las marcas del mapa y se incluirá en el EKF. El tiempo de cómputo del EKF presenta una complejidad de $O(n^2)$, siendo n el número de marcas presentes en el filtro.

El mapa esta formado por una serie de **marcas naturales**. Estas marcas se identificarán por sus correspondientes *apariencias* una vez capturadas por la cámara. A medida que la cámara se mueve por el entorno visitando nuevos lugares, se incorporan nuevas marcas al vector de estado incrementando el tamaño del mapa 3D del entorno. Uno de las principales requisitos de estas marcas, es que sean **estáticas**, ya que en caso contrario si se seleccionan muchas marcas dinámicas, la calidad del mapa puede verse reducida drásticamente y el proceso del EKF puede llegar a divergir.

Estas **marcas naturales** no son simplemente el resultado del proceso de mapeado, sino que además son el medio a través del cuál la cámara es capaz de localizarse a sí misma. Este proceso se basa en el uso de un modelo de proyección inverso, de forma que teniendo las coordenadas de proyección (pixélicas) de cada marca en ambas cámaras, es posible calcular la posición 3D respecto del sistema de referencia global. Obteniendo la posición absoluta de diferentes marcas, es posible deducir la posición y orientación de la propia cámara respecto del entorno. En la figura 6.1 se puede observar a modo de ejemplo el proceso de captura de marcas naturales.

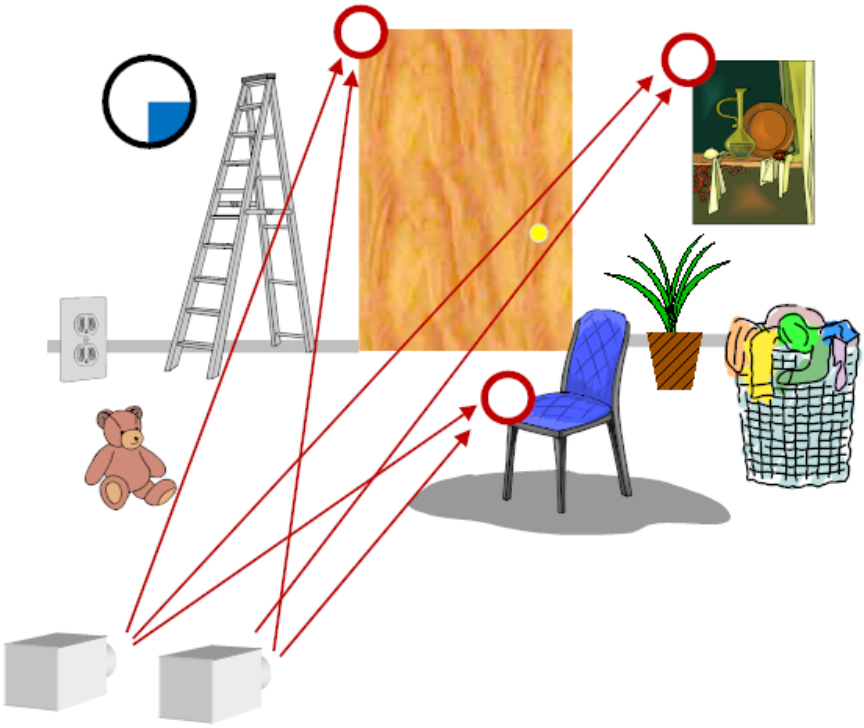


Figura 6.1: Proceso de captura de marcas naturales

6.1. Vector de Estado

El primer paso a la hora de definir los elementos que componen el EKF, será la descripción del vector de estado. Dada la estructura del sistema, el vector de estado se divide en dos partes. Por un lado, tenemos el **vector de estado correspondiente a la cámara izquierda** que esta compuesto por:

$$\mathbf{X_{v\ [13,1]}} = (x_{cam}\ y_{cam}\ z_{cam},\ q_0\ q_x\ q_y\ q_z,\ v_x\ v_y\ v_z,\ \omega_x\ \omega_y\ \omega_z)^t \tag{6.1}$$

A partir del vector de estado de la ecuación 6.1, se pueden identificar las siguientes componentes:

- La **posición** absoluta de la cámara izquierda en el sistema de referencia global:

$$\mathbf{X_{cam\ [3,1]}} = (x_{cam}\ y_{cam}\ z_{cam})^t \tag{6.2}$$

- La **orientación** de la cámara izquierda con respecto al sistema de referencia global definida mediante un cuaternión:

$$\mathbf{q_{cam\ [4,1]}} = (q_0\ q_x\ q_y\ q_z)^t \tag{6.3}$$

- La información referente a la posición y la orientación de la cámara izquierda respecto del sistema de referencia global, se pueden agrupar en nuevo vector $\mathbf{X_p}$

$$\mathbf{X_{p\ [7,1]}} = (x_{cam}\ y_{cam}\ z_{cam},\ q_0\ q_x\ q_y\ q_z)^t \tag{6.4}$$

- La **velocidad lineal** de la cámara izquierda con respecto al sistema de referencia global:

$$\mathbf{v_{cam\ [3,1]}} = (v_x\ v_y\ v_z)^t \tag{6.5}$$

- La **velocidad angular** de la cámara izquierda con respecto al sistema de referencia global:

$$\omega_{\mathbf{cam\ [3,1]}} = (\omega_x\ \omega_y\ \omega_z)^t \tag{6.6}$$

Se utiliza un cuaternión para describir la rotación de la cámara izquierda con respecto al entorno q_{cam}. El uso de cuaterniones está muy extendido en el campo de la robótica, ya que es sencillo realizar rotaciones concatenadas secuencialmente, además de existir una relación directa entre los valores de las componentes del cuaternión y la matriz de rotación R, que el cuaternión representa. Para más información sobre las propiedades de los cuaterniones se puede consultar el apéndice A.

En el vector de estado se incluyen también la velocidad lineal y la velocidad angular de la cámara izquierda. Esto es debido al modelo de movimiento utilizado. Este modelo de movimiento se explica en mayor detalle en la sección 6.4.

Por otro lado, se definen los **vectores de estado estimados de las diferentes marcas que forman el mapa completo** utilizadas en el filtro. De acuerdo con la distinción anterior, se utilizan dos tipos de parametrizaciones de las marcas, **parametrización 3D** y **parametrización inversa**:

- **Parametrización 3D**: En esta parametrización, el vector de estado de la marca contiene la información sobre la posición absoluta de dicha marca respecto del sistema de referencia global.

$$\mathbf{Y_{i\ 3D\ [3,1]}} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{6.7}$$

- **Parametrización Inversa**: En esta parametrización, el vector de estado de la marca contiene la información sobre la posición inicial 3D en el que la marca fue vista por primera vez con respecto al sistema de referencia global (X_{ori}), los ángulos en azimuth y en elevación de la marca con respecto a la cámara izquierda (θ, ϕ), y la inversa de la profundidad ($1/\rho$).

$$\mathbf{Y_{i\ INV\ [6,1]}} = \begin{pmatrix} X_{ori} \\ \theta \\ \phi \\ 1\ /\ \rho \end{pmatrix} \tag{6.8}$$

La parametrización inversa de las marcas será explicada con mayor profundidad en la sección 6.3, así como la decisión de cuando una marca se parametriza con una parametrización 3D o inversa.

Finalmente, agrupando todos los elementos anteriores, el **vector de estado global** se define como sigue:

$$\mathbf{X} = \begin{pmatrix} X_v \\ Y_{1\ 3D} \\ \vdots \\ Y_{n\ 3D} \\ Y_{1\ INV} \\ \vdots \\ Y_{m\ INV} \end{pmatrix} \tag{6.9}$$

$$\mathbf{P} = \begin{pmatrix} P_{X_vX_v} & P_{X_vY_{1\ 3D}} & \cdots & P_{X_vY_{n\ 3D}} & P_{X_vY_{1\ INV}} & \cdots & P_{X_vY_{m\ INV}} \\ P_{Y_{1\ 3D}X_v} & P_{Y_{1\ 3D}Y_{1\ 3D}} & \cdots & P_{Y_{1\ 3D}Y_{n\ 3D}} & P_{Y_{1\ 3D}Y_{1\ INV}} & \cdots & P_{Y_{1\ 3D}Y_{m\ INV}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ P_{Y_{n\ 3D}X_v} & P_{Y_{n\ 3D}Y_{n\ 3D}} & \cdots & P_{Y_{n\ 3D}Y_{n\ 3D}} & P_{Y_{n\ 3D}Y_{1\ INV}} & \cdots & P_{Y_{n\ 3D}Y_{m\ INV}} \\ P_{Y_{1\ INV}X_v} & P_{Y_{1\ INV}Y_{1\ 3D}} & \cdots & P_{Y_{1\ INV}Y_{n\ 3D}} & P_{Y_{1\ INV}Y_{1\ INV}} & \cdots & P_{Y_{1\ INV}Y_{m\ INV}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ P_{Y_{n\ INV}X_v} & P_{Y_{n\ INV}Y_{n\ 3D}} & \cdots & P_{Y_{n\ INV}Y_{n\ 3D}} & P_{Y_{n\ INV}Y_{1\ INV}} & \cdots & P_{Y_{n\ INV}Y_{m\ INV}} \end{pmatrix} \tag{6.10}$$

En las ecuaciones 6.9 y 6.10, **n** representa el número de marcas con parametrización 3D presentes en el mapa, mientras que **m** representa por su parte el número de marcas con parametrización inversa presentes en el mapa.

La dimensión del vector de estado **X** es de $(13 + n \cdot 3 + m \cdot 7) \times 1$, mientras que la dimensión de la matriz de covarianza **P** es una matriz cuadrada de $(13 + n \cdot 3 + m \cdot 7) \times (13 + n \cdot 3 + m \cdot 7)$.

6.2. Filtro Extendido de Kalman (EKF)

Se consigue estimar la posición y la orientación de la cámara en cada instante de tiempo mediante el uso del EKF. El Filtro de Kalman es una poderosa herramienta para sistemas lineales, pero dado el caso del presente trabajo, en el que el sistema a modelar es la posición y orientación de un par estéreo movido con la mano, el sistema es claramente no lineal. Dado que no se puede suponer una función de estimación del próximo estado $f(X)$ lineal, el EKF proporciona la mencionada estimación linealizando $f(X)$ en cada instante de tiempo, siendo esta la principal diferencia con respecto al filtro de Kalman.

De manera general, el EKF presenta las etapas que se observan en la figura 6.2. A continuación, se describe la implementación del EKF en dónde k representa el índice temporal.

- **Fase de Predicción**: El primer paso del filtro consiste en predecir el vector de estado en el próximo instante de tiempo. Para ello se utiliza la mencionada función $f(X)$ la cuál será explicada en detalle en la sección 6.4. Las ecuaciones de predicción son las siguientes:

$$\hat{\mathbf{X}}(\mathbf{k+1}|\mathbf{k}) = f(X(k|k)) \tag{6.11}$$

$$\hat{\mathbf{P}}(\mathbf{k+1}|\mathbf{k}) = \frac{\partial f}{\partial X}(k|k) \cdot P(k|k) \cdot \left(\frac{\partial f}{\partial X}(k|k)\right)^t + Q(k) \tag{6.12}$$

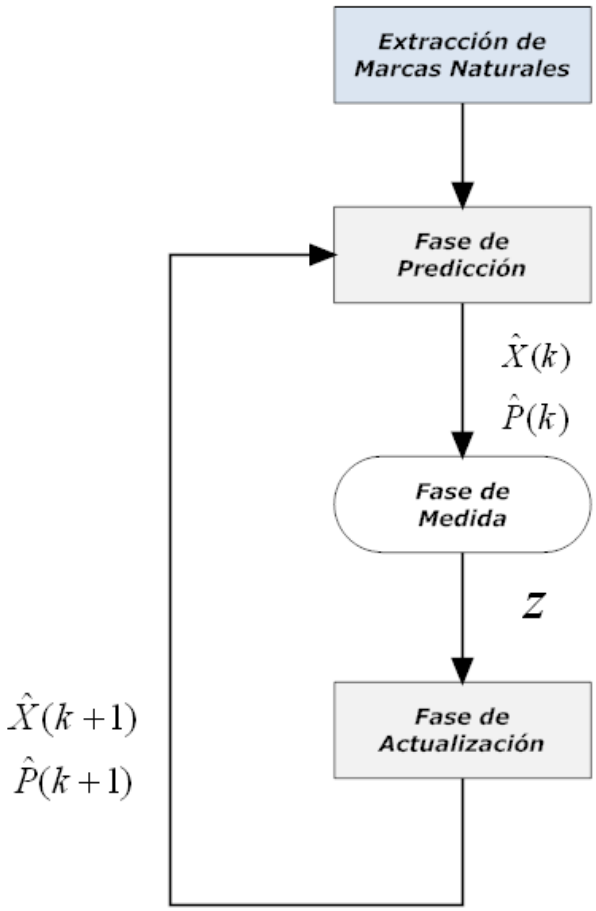


Figura 6.2: Esquema fases del EKF

A partir de la ecuación 6.11, se obtiene directamente la **predicción del vector de estado** para el próximo instante de tiempo $k+1$. La ecuación 6.12 permite obtener la **predicción de la matriz de covarianza** en el próximo instante de tiempo. En esta última ecuación, la matriz $Q(k)$ representa la **covarianza de ruido de proceso** para el instante de tiempo k. La obtención de esta covarianza de ruido de proceso se explica en la sección 6.4.

- **Fase de Actualización**: Una vez completada la fase de predicción, el siguiente paso consiste en realizar las medidas correspondientes y actualizar el filtro con la información obtenida. Para ello se utilizan las siguientes ecuaciones:

$$\hat{\mathbf{X}}\left(\mathbf{k+1}|\mathbf{k+1}\right) = \hat{X}\left(k+1|k\right) + W(k+1)\cdot\eta\left(k+1\right)_{tot} \tag{6.13}$$

$$\mathbf{P}\left(\mathbf{k+1}|\mathbf{k+1}\right) = P\left(k+1|k\right) - W\left(k+1\right)\cdot S\left(k+1\right)\cdot\left(W\left(k+1\right)\right)^{t} \tag{6.14}$$

En este caso, la ecuación 6.13 proporciona el valor del vector de estado una vez actualizado con las medidas realizadas. En esta misma ecuación, el parámetro η_{tot} se corresponde con el **vector de innovación**, que es la diferencia entre el vector de medida z_{tot} y el vector de predicción de dichas medidas h_{tot}. Para poder actualizar el vector de estado, será necesario multiplicar η_{tot} por la matriz de ganancias W, la cuál se explicará en la sección 6.5.4.

La ecuación 6.14 permite obtener el valor actualizado de la matriz de covarianza del vector de estado total. Para ello, es necesario calcular la matriz S que representa la **covarianza de ruido de medida** (covarianza del vector de innovación), y multiplicar dicha matriz por la matriz de ganancias W.

Una vez finalizada la fase de actualización, se avanza el filtro pasando a la siguiente iteración y al siguiente instante de tiempo.

6.3. Parametrización Inversa de las Marcas

6.3.1. Consideraciones Previas

Una cámara estéreo puede proporcionar una estimación precisa en profundidad de los puntos hasta un cierto rango de distancia, determinado principalmente por la línea de base existente entre ambas cámaras. Por lo tanto, se pueden distinguir dos regiones: una cercana a la cámara en la cuál el sensor es capaz de estimar de manera precisa la profundidad de los puntos, y una zona lejana en la cuál el sensor estéreo se aproxima a una cámara monocular, perdiendo la precisión en la estimación de la profundidad, pero ofreciendo una mejor estimación angular.

Un **sistema de visión estéreo ideal** consta de dos cámaras perfectamente alineadas, es decir con sus ejes ópticos paralelos. Suponiendo resuelto el problema de correspondencia de un punto para la cámara derecha y para la cámara izquierda, para calcular la profundidad Z a la que se encuentra el punto P a partir de las proyecciones p_L y p_R es necesario conocer previamente la distancia focal f, las coordenadas de los centros ópticos c_L y c_R, y la distancia entre los centros de proyección de ambas cámaras B. Como se puede observar en la figura 6.3 se obtienen dos triángulos formados por los puntos (p_L, P, p_R) y (O_L, P, O_R), lo que implica la relación:

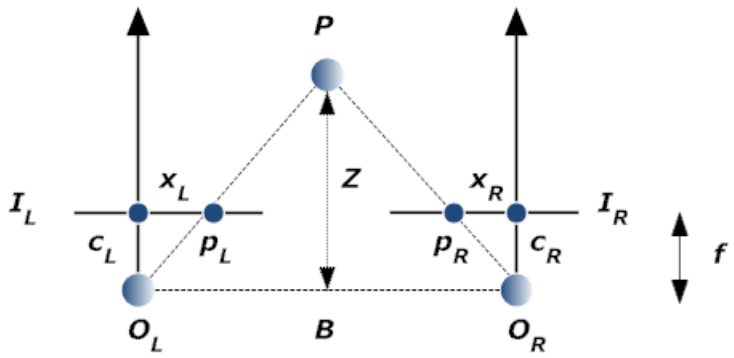


Figura 6.3: Profundidad estimada a partir de la disparidad entre puntos correspondientes

$$\frac{B + x_L - x_R}{Z - f} = \frac{B}{Z} \Rightarrow Z = f \cdot \frac{B}{d} \tag{6.15}$$

donde $d = x_R - x_L$ es la denominada **disparidad horizontal** que mide la diferencia en el eje x del plano imagen de los puntos correspondientes a las cámaras derecha e izquierda. Como se puede observar, la profundidad de los objetos es inversamente proporcional a la disparidad. Para obtener una disparidad pixélica, se realiza un cambio de coordenadas métricas a pixélicas (ver ecuación 4.8), y suponiendo los mismos parámetros intrínsecos para ambas cámaras se obtiene:

$$Z = f \cdot \frac{B}{x_R - x_L} = f \cdot \frac{B}{(u_R - u_0)\, d_x - (u_L - u_0)\, d_x} = \frac{f}{d_x} \cdot \frac{B}{u_R - u_L} = f_x \cdot \frac{B}{d_u} \tag{6.16}$$

Procediendo de una manera similar [35], se pueden obtener las expresiones para las coordenadas 3D X e Y, obteniendo las siguientes ecuaciones:

$$X = \frac{B \cdot (u_L - u_0)}{d_u} \tag{6.17}$$

$$Y = \frac{B \cdot (v_L - v_0)}{d_u} \tag{6.18}$$

Aunque se trate de un modelo simplificado, a partir de la ecuación 6.16 se pueden extraer conclusiones aproximadas acerca de las características de rango y precisión en la medida de distancia de un sistema de visión estéreo. Dada la distancia entre los centros ópticos de las cámaras B, la distancia focal pixélica para el eje x f_x y la resolución pixélica de las imágenes (W, H), se puede calcular la precisión en la medida de distancia desde la máxima disparidad $d_u = W - 1$ (mínima distancia) hasta la mínima disparidad $d_u = 1$ (máxima profundidad) a partir de la siguiente expresión:

$$\Delta Z_i = Z_i - Z_{i-1} = f_x \cdot B \left(\frac{1}{d_{u_i} - 1} - \frac{1}{d_{u_i}} \right) = f_x \cdot B \cdot \frac{1}{d_{u_i}^2 - d_{u_i}} \tag{6.19}$$

En la figura 6.4 se puede observar la relación entre la distancia entre cámaras B y el máximo rango de distancia teórico, para diferentes valores de distancia focal en el eje x f_x. En esta figura, se muestra la relación entre la profundidad y su precisión, para distancias entre cámaras desde $15cm$ hasta $40cm$ de imágenes con resolución de 320×240 y de distancia focal $f_x = 203{,}16$. A medida que la profundidad de los objetos aumenta, la precisión disminuye considerablemente, siendo menor el efecto de la degradación en la medida según sea mayor la distancia entre cámaras B. Por ejemplo, para objetos situados a una distancia Z de $14m$, una distancia entre cámaras B de $15cm$ tiene aproximadamente un error de $\pm 12m$, mientras que para una distancia entre cámaras B de $40cm$ el error en la medida es de aproximadamente $\pm 3m$.

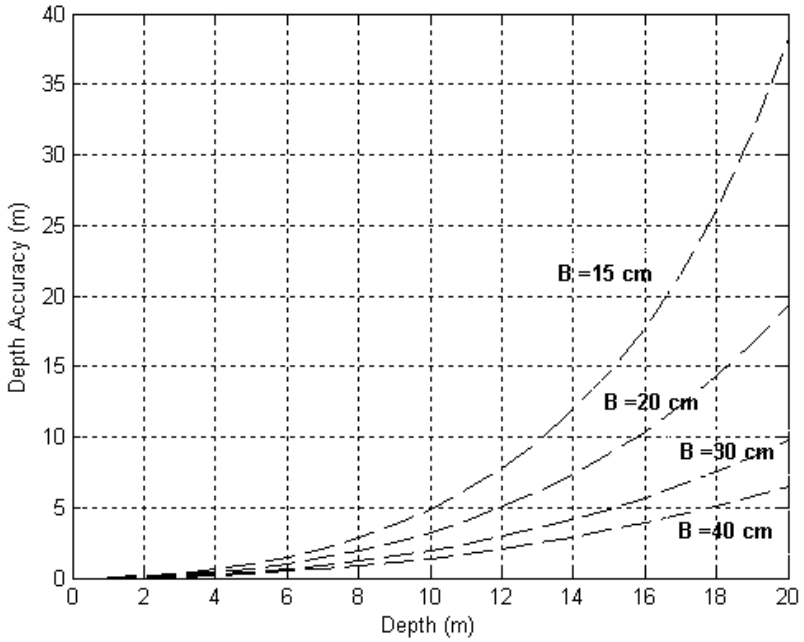


Figura 6.4: Relación entre la profundidad y precisión en la medida

La conclusión principal que se obtiene del análisis anterior, es que para obtener más precisión en las medidas de profundidad de los objetos a partir de un sistema de visión estéreo, interesa

que por un lado, las cámaras estén lo más alejado posible entre sí, y por otro que la resolución pixélica sea lo más grande posible.

Sin embargo, y dado que se trata de una aplicación para asistencia a personas invidentes y que debe funcionar en tiempo real, la línea de base de las cámaras B no debe ser excesivamente grande, del mismo modo que la resolución de las imágenes, ya que si no el tiempo de cómputo podría superar las restricciones de tiempo real. Por lo tanto en el presente trabajo estos parámetros vienen en parte impuestos por las propias condiciones de la aplicación, la línea de base B es de $15cm$ y la resolución de las imágenes es de 320×240.

Debido al alto error en la medida de la profundidad existente para una línea de base de $15cm$, se propone realizar una distinción para marcas cercanas a la cámara (parametrización 3D) y para marcas lejanas (parametrización inversa), de tal modo que la parametrización 3D proporcione medidas fiables de profundidad y la parametrización inversa proporciona medidas fiables angulares.

6.3.2. Formulación de la Parametrización Inversa de las Marcas

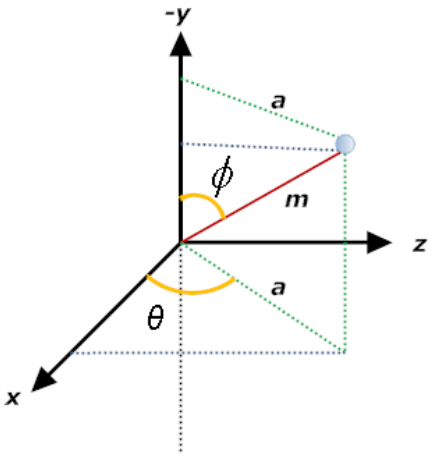


Figura 6.5: Obtención del vector unitario m

Dado un punto en el espacio 3D correspondiente al sistema de referencia global. Para obtener la expresión del vector unitario m, cuya dirección viene determinada por el centro de coordenadas del sistema de referencia de la cámara izquierda y el punto 3D en dónde se encuentra la marca en el espacio, es necesario realizar el desarrollo que se expone a continuación.

Sea un punto en el espacio 3D definido por (x_i, y_i, z_i), este punto se puede expresar de acuerdo con el sistema de coordenadas de la figura 6.5 a partir de las siguientes relaciones:

$$\begin{cases} x_i &= a \cdot \cos\theta_i \\ y_i &= -\left|\hat{M}\right| \cdot \cos\phi_i \\ z_i &= a \cdot \sin\theta_i \end{cases} \tag{6.20}$$

La relación entre la proyección a y el vector $\hat{M}$ viene dada por:

$$a = \left|\hat{M}\right| \cdot \sin\phi_i \tag{6.21}$$

A partir de las relaciones anteriores, podemos obtener la relación entre el vector $\hat{M}$ y el punto (x_i, y_i, z_i) como sigue:

$$\hat{M} = x_i \cdot \hat{i} + y_i \cdot \hat{j} + z_i \cdot \hat{k} = \left|\hat{M}\right| \sin\phi_i \cos\theta_i \cdot \hat{i} - \left|\hat{M}\right| \cos\phi_i \cdot \hat{j} + \left|\hat{M}\right| \sin\phi_i \sin\theta_i \cdot \hat{k} \quad (6.22)$$

Si se normaliza el vector $\hat{M}$, obtenemos el vector unitario m cuyas componentes son:

$$\mathbf{m}_{[3,1]} = \frac{\hat{M}}{\left|\hat{M}\right|} = (\sin\phi_i \cos\theta_i, -\cos\phi_i, \sin\phi_i \sin\theta_i)^t \quad (6.23)$$

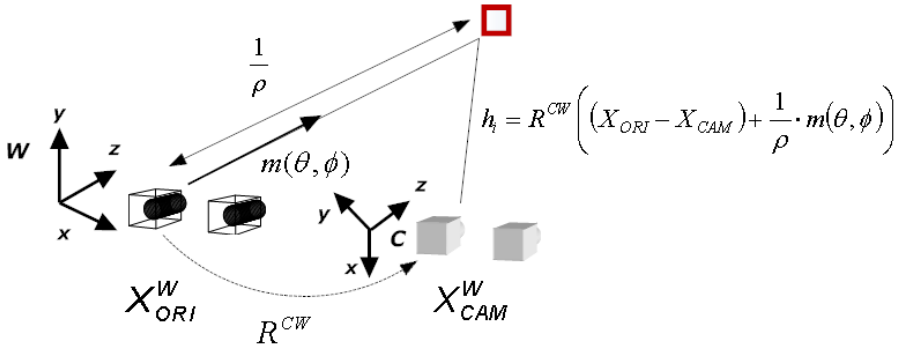


Figura 6.6: Parametrización inversa de la marca

El valor de los ángulos de azimuth y elevación se calcula a partir de las siguientes ecuaciones:

$$\theta_\mathbf{i} = tan^{-1}\left(\frac{z}{x}\right) \quad (6.24)$$

$$\phi_\mathbf{i} = tan^{-1}\left(\frac{\sqrt{x^2 + z^2}}{y}\right) \quad (6.25)$$

$$\frac{1}{\rho_i} = d_i \quad (6.26)$$

6.3.3. Elección entre Parametrización 3D o Inversa

Una vez que se han definido ambas parametrizaciones, el problema a resolver, es encontrar el valor de disparidad d_u o de distancia Z para el cuál, una parametrización determinada presenta resultados más precisos en la estimación de la medida 3D o angular. Cuanto más lineal sea una medida, mejores estimaciones se obtendrán con el Filtro de Kalman.

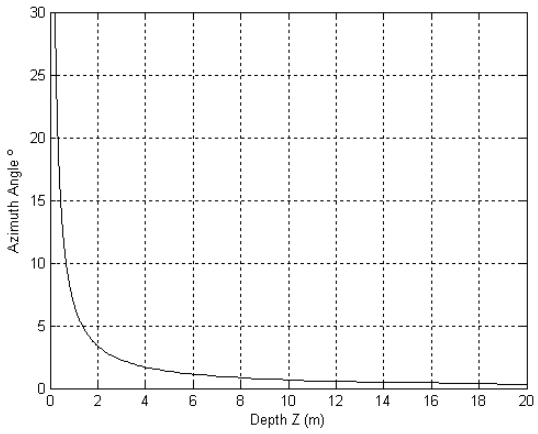
Observando la figura 6.4, para una línea de base de 15 cm, un valor de compromiso para elegir entre una parametrización u otra podría ser una distancia $Z = 10m$ para el cuál podríamos obtener un error aproximado de $\pm 5m$ al cometer un error de disparidad de ± 1 pixel. Para distancias mayores, el error cometido es mayor que el 50 % lo cuál no es aceptable para nuestro sistema de medida.

Considerando las ecuaciones 6.17, 6.18, podemos obtener el valor de las coordenadas X, Y máximas $(u_L - u_0 = W/2)$, $(v_L - v_0 = H/2)$ que se obtienen considerando una disparidad horizontal máxima $d_u = W$:

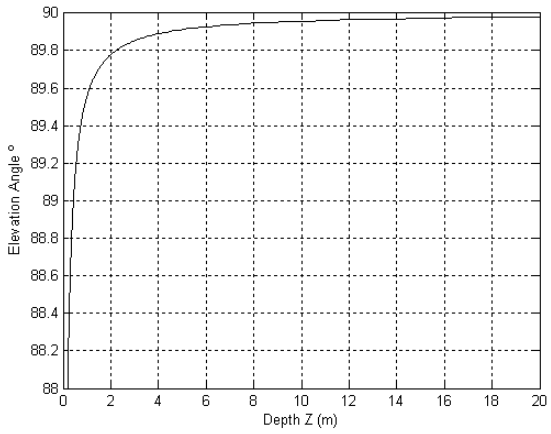
$$X = \frac{B \cdot (W/2)}{W} = \frac{B}{2} \tag{6.27}$$

$$Y = \frac{B \cdot (H/2)}{W} = \frac{B}{2} \cdot \frac{H}{W} \tag{6.28}$$

Para los valores X,Y obtenidos de las ecuaciones 6.27, 6.28 representamos el valor de los ángulos θ, ϕ para distintos valores de distancia Z (ver figura 6.3.3).



(a) Ángulo de azimuth vs profundidad



(b) Ángulo de elevación vs profundidad

Como se puede observar en la figura anterior, la medida de los ángulos de azimuth y elevación, es más lineal a medida que aumenta la distancia Z. Esto se contrapone a la figura 6.4 en la cuál, el error en Z aumenta a medida que aumenta Z o disminuye la disparidad horizontal d_u.

Para poder obtener para un umbral de distancia Z que nos indique que parametrización proporciona mejores resultados, se realiza un estudio de la no linearidad de la distancia Z en función de la disparidad horizontal d_u y de los ángulos θ y ϕ en función de la distancia Z.

Para ello consideremos el desarrollo en serie de Taylor para la primera derivada de una función continua f que depende de una variable Z:

$$\frac{\partial f}{\partial Z}(z + \Delta z) \approx \left.\frac{\partial f}{\partial Z}\right|_z + \left.\frac{\partial^2 f}{\partial Z^2}\right|_z \Delta z \tag{6.29}$$

Si consideramos el cociente entre la segunda y la derivada primera de f multiplicado por el factor Δz, obtenemos la expresión de un índice de linearidad adimensiónal de la función f en relación con la variable Z:

$$\mathbf{L_f} = \left| \frac{\frac{\partial^2 f}{\partial Z^2} \cdot \Delta Z}{\frac{\partial f}{\partial Z}} \right| \tag{6.30}$$

Si obviamos en la ecuación 6.30 los valores absolutos, podemos expresar la ecuación 6.29 en función del índice de linearidad L_f como sigue:

$$\frac{\partial f}{\partial Z}(z+\Delta z) \approx \left.\frac{\partial f}{\partial Z}\right|_z \cdot (1+L_f) \tag{6.31}$$

Atendiendo a la expresión de la ecuación 6.31 podemos sacar varias conclusiones:

- Si el índice de linearidad L_f es igual a cero para un punto Z_i, esto quiere decir que la función f es lineal con respecto a la variable Z en un intervalo ΔZ centrado sobre el punto Z_i. Si la segunda derivada es igual a cero, esto implica que f es una función lineal de la variable Z ya que su primera derivada es una constante y la segunda derivada es nula.
- Si el índice de linearidad L_f toma valores distintos de cero, esto quiere decir que en la aproximación de Taylor de la primera derivada, es necesario incluir el factor correspondiente a la segunda derivada, siendo por lo tanto este factor no despreciable y distinto de cero, lo que implica que la función f no es lineal con la variable Z para ese intervalo ΔZ.

Considerando el desarrollo anterior, a continuación se obtendrán índices de linearidad para la profundidad Z y para los ángulos de azimuth y elevación θ y ϕ respectivamente.

6.3.3.1. Linearidad de la Profundidad Z

Se propone comparar el valor de la derivada segunda para un valor de disparidad horizontal determinado d_{u_i}, con el valor de la derivada primera para ese valor de disparidad, obteniendo el valor de un índice adimensional que representa la desviación con respecto a la linearidad.

$$\mathbf{L_Z} = \left| \frac{\frac{\partial^2 Z}{\partial d_{u_i}^2} \cdot \Delta d_u}{\frac{\partial Z}{\partial d_{u_i}}} \right| \tag{6.32}$$

En la figura 6.7 se puede observar una figura de la distancia Z en función de la disparidad horizontal d_{u_i}, para una línea de base $B = 15cm$. Como se puede apreciar, para valores de disparidad horizontal altos, esto implica valores de distancias Z bajas, siendo la relación entre ambas variables es más lineal. Mientras que para valores más elevados de profundidad, se observa que la relación con la disparidad horizontal deja de ser lineal.

Las expresiones de la derivada primera y la derivada segunda de la distancia Z con respecto la disparidad horizontal d_{u_i} son las siguientes respectivamente:

$$\frac{\partial Z}{\partial d_u} = -\frac{f_x \cdot B}{d_u^2} \tag{6.33}$$

$$\frac{\partial^2 Z}{\partial d_u^2} = \frac{2 \cdot f_x \cdot B}{d_u^3} \tag{6.34}$$

Teniendo en cuenta la relación existente entre la profundidad Z y la disparidad horizontal d_u podemos obtener el índice de linearidad de la profundidad en función de la misma variable de profundidad Z atendiendo a la expresión de la ecuación 6.16:

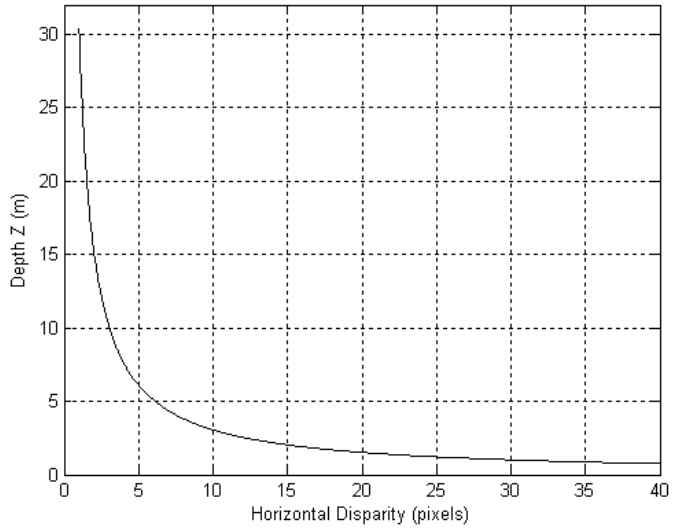


Figura 6.7: Profundidad en función de la disparidad

$$\mathbf{L_z} = \frac{2 \cdot \Delta d_u}{d_u} = \frac{2 \cdot Z \cdot \Delta d_u}{f_x \cdot B} \tag{6.35}$$

En la figura 6.8 se representa el índice de linearidad obtenido L_Z considerando un valor de $\Delta d_u = 1 pixel$. Como se puede observar, la principal conclusión, es que para valores de profundidad elevados, el índice de linearidad crece proporcionalmente con la profundidad. Esta conclusión coincide con la figura 6.4 en la cuál se observa como el error en la precisión aumenta a medida que aumenta la profundidad, y esto es debido precisamente a la no linealidad de la profundidad para distancias elevadas.

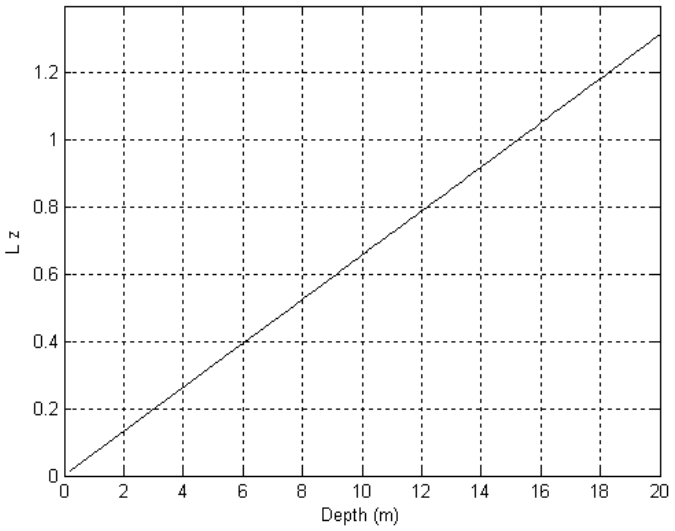


Figura 6.8: Indice de linearidad para la profundidad Z

6.3.3.2. Linearidad de los Ángulos de Azimuth y Elevación

Para poder obtener la linearidad de los ángulos de azimuth y elevación, se propone el cálculo del siguiente índice de linearidad angular L_a:

$$\mathbf{L_a} = \left| \frac{\frac{\partial^2 \theta_i}{\partial z^2} \cdot \Delta z}{\frac{\partial \theta_i}{\partial z}} \right| + \left| \frac{\frac{\partial^2 \phi_i}{\partial z^2} \cdot \Delta z}{\frac{\partial \phi_i}{\partial z}} \right| \tag{6.36}$$

Las expresiones de las derivadas primera y segunda del ángulo de azimuth θ con respecto a la distancia Z son respectivamente:

$$\frac{\partial \theta_i}{\partial z} = \frac{1}{x\left(1 + \frac{z^2}{x^2}\right)} \tag{6.37}$$

$$\frac{\partial^2 \theta_i}{\partial z^2} = -\frac{2 \cdot z}{x^3\left(1 + \frac{z^2}{x^2}\right)^2} \tag{6.38}$$

A partir de las ecuaciones 6.37 y 6.38, podemos obtener el índice de linearidad para el ángulo de azimuth en función de la profundidad:

$$\mathbf{L}_\theta = \frac{2 \cdot z}{x^2\left(1 + \frac{z^2}{x^2}\right)} \cdot \Delta z \tag{6.39}$$

Las expresiones de las derivadas primera y segunda del ángulo de elevación ϕ con respecto a la distancia Z son respectivamente:

$$\frac{\partial \phi_i}{\partial z} = \frac{z}{y\sqrt{x^2 + y^2} \cdot \left(1 + \frac{x^2 + z^2}{y^2}\right)} \tag{6.40}$$

$$\frac{\partial^2 \phi_i}{\partial z^2} = \frac{y\left(x^4 - 2 \cdot z^4 + x^2\left(y^2 - z^2\right)\right)}{\left(x^2 + z^2\right)^{3/2}\left(x^2 + y^2 + z^2\right)^2} \tag{6.41}$$

Del mismo modo que para el ángulo de azimuth, a partir de las ecuaciones 6.40 y 6.41, podemos obtener el índice de linearidad para el ángulo de elevación en función de la profundidad:

$$\mathbf{L}_\phi = \frac{x^4 - 2 \cdot z^4 + x^2\left(y^2 - z^2\right)}{z\left(x^2 + z^2\right)\left(x^2 + y^2 + z^2\right)} \cdot \Delta z \tag{6.42}$$

Como se puede observar en las ecuaciones 6.39,6.42, el cálculo del índice de linearidad angular depende de la posición 3D (x, y, z) de un punto en el espacio. Para poder obtener un valor de índice de linearidad angular se ha realizado una simulación considerando varios valores de puntos 3D, y se ha obtenido un índice de linearidad medio. En la figura 6.9 se representa el índice de linearidad obtenido L_a para distintos valores del parámetro Δz:

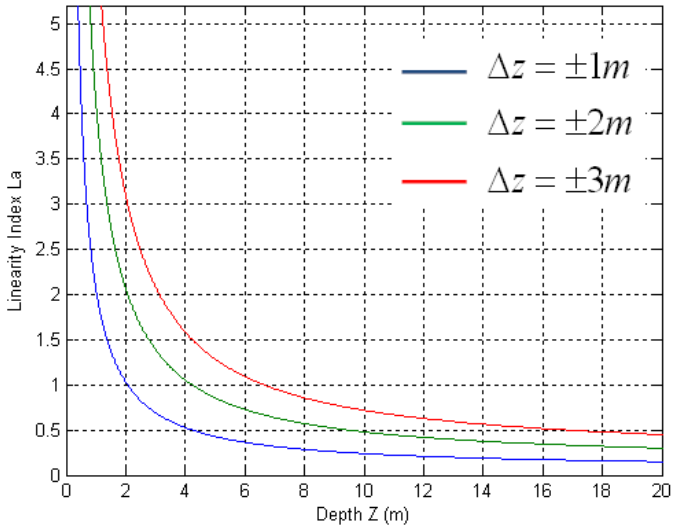


Figura 6.9: Indice de linearidad para los ángulos de azimuth y elevación

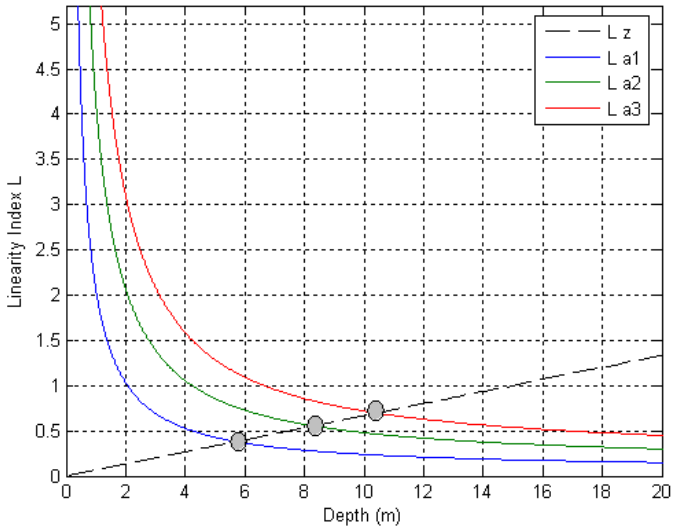


Figura 6.10: Punto de corte entre los índices de linearidad

6.3.3.3. Elección de Umbral de Profundidad

Si se representan las curvas del índice de linearidad de profundidad L_z y el índice de linearidad angular L_a se puede observar que ambas curvas se cortan en un punto para una profundidad Z.

Para cada una de las curvas del índice de linearidad angular L_{a_i} el punto de corte con la curva del índice de linearidad de profundidad L_z, z_i resulta ser:

- L_{a_1} , $\Delta z = \pm 1m \Longrightarrow z_1 \ = \ 5{,}71\ m$

- L_{a_2} , $\Delta z = \pm 2m \Longrightarrow z_2 \ = \ 8{,}48\ m$

- L_{a_3} , $\Delta z = \pm 3m \Longrightarrow z_3 \ = \ 10{,}43\ m$

Analizando la figura 6.10 se puede observar como para valores de profundidad Z superiores al valor del punto de corte, resulta ser más lineal la medida angular, mediante que para valores de profundidad Z inferiores al valor del punto de corte, la medida de profundidad resulta ser más lineal. Por lo tanto, como valor umbral para parametrizar las marcas como 3D o como marcas inversas, en el presente trabajo se ha elegido el valor de distancia Z de 5,71 m

6.4. Modelo de Predicción (Movimiento)

En esta sección se define el modelo de movimiento necesario para la implementación del EKF. Los dos elementos principales del EKF son el vector de estado X y la matriz de covarianza P. En la fase de predicción del filtro, el objetivo es estimar ambos elementos en el instante siguiente de tiempo, para lo cuál es necesario el uso de un modelo de movimiento que se adapte lo mejor posible a la dinámica del sistema.

En este trabajo, el objeto a modelas es el movimiento de una cámara estéreo movida con la mano. Por lo tanto, no se dispone de información a priori sobre cómo va a ser dicho movimiento. El modelo de movimiento implementado supone una *velocidad lineal y angular constante* en cada estado. Esto no significa que se asuma que la cámara se mueva a una velocidad constante durante todo el tiempo, sino que el modelo estadístico del movimiento de la cámara en un instante de tiempo se esperan aceleraciones aleatorias indeterminadas con un perfil gaussiano. Una de las consecuencias de este modelo, es que se impone un *cierto suavizado* al movimiento de la cámara, ya que se supone que aceleraciones muy bruscas son relativamente poco probables. En la figura 6.11 se puede observar un esquema de este tipo de movimiento.

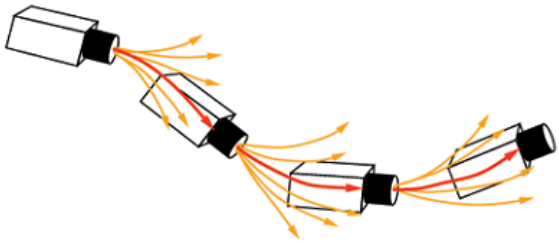


Figura 6.11: Modelo de movimiento suavizado

Para predecir el vector de estado en el instante de tiempo siguiente, se define la función $f_v\left(X_v(k|k)\right)$, la cual proporciona como resultado $X_v(k+1|k)$ la predicción del estado de la cámara . Dado que una de las restricciones de las marcas es que sean estáticas respecto al entorno, la predicción de su posición coincidirá con la posición en el estado actual. De esta forma se puede definir la predicción del estado total de la siguiente forma:

$$\mathbf{f} = \hat{\mathbf{X}}(\mathbf{k+1}|\mathbf{k}) = \begin{pmatrix} f_v\left(X_v(k|k)\right) \\ \hat{Y}_{1\ 3D}(k|k) \\ \vdots \\ \hat{Y}_{n\ 3D}(k|k) \\ \hat{Y}_{1\ INV}(k|k) \\ \vdots \\ \hat{Y}_{m\ INV}(k|k) \end{pmatrix} = \begin{pmatrix} f_v\left(X_v(k|k)\right) \\ Y_{1\ 3D}(k|k) \\ \vdots \\ Y_{n\ 3D}(k|k) \\ Y_{1\ INV}(k|k) \\ \vdots \\ Y_{m\ INV}(k|k) \end{pmatrix} \tag{6.43}$$

Para modelar el **paso de un estado a otro**, se supone **un cambio de aceleración lineal y angular aleatorios de media 0 y con distribución gaussiana**. Por lo tanto, se define un vector aleatorio en el que se definen los cambios de velocidad (lineal y angular) de un estado a otro:

$$\mathbf{n} = \begin{pmatrix} V \\ \Omega \end{pmatrix} = \begin{pmatrix} a \cdot \Delta t \\ \alpha \cdot \Delta t \end{pmatrix} \tag{6.44}$$

Dependiendo de las circunstancias V y Ω podrían estar acopladas, por ejemplo, con algún tipo de movimiento que produzca cambios correlados en ambas velocidades al mismo tiempo. Sin embargo, si se consideran a y α independientes, se puede formar la matriz de covarianza del vector n como:

$$\mathbf{P_n} = \begin{pmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\Omega^2 \end{pmatrix} \tag{6.45}$$

A partir de las suposiciones anteriores se obtiene la función $f_v\left(X_v(k|k)\right)$:

$$\mathbf{f_v} = \begin{pmatrix} X_{cam} + (v_{cam} + V) \cdot \Delta t \\ q_{cam} \times q\left[(\omega + \Omega) \cdot \Delta t\right] \\ v_{cam} + V \\ \omega + \Omega \end{pmatrix} \tag{6.46}$$

Como se ha comentado anteriormente, para poder obtener la predicción de la covarianza total P, es necesario obtener el **ruido de predicción** Q_v. Este ruido se calcula a partir de la siguiente operación con jacobianos:

$$\mathbf{Q_v} = \frac{\partial f_v}{\partial n} P_n \left(\frac{\partial f_v}{\partial n}\right)^t \tag{6.47}$$

La tasa de crecimiento en la incertidumbre de este modelo de movimiento esta determinada por la matriz P_n y dependiendo de los valores de los parámetros de dicha matriz, se define la *suavidad* del movimiento esperado. Para pequeños valores de P_n se espera un movimiento muy suave con pequeñas aceleraciones y el modelo se adaptaría bien para movimientos de este tipo, pero sería incapaz de manejar cambios bruscos de movimiento. Por el contrario, valores altos de P_n significa que la incertidumbre en el sistema crece de manera significativa a cada instante de tiempo, y aunque esto permite manejar rápidas aceleraciones, también implica que se deben de tomar en cada instante de tiempo muchas medidas correctas.

La implementación del EKF requiere también del cálculo del jacobiano $\frac{\partial f}{\partial X}$ y del **ruido de predicción total** Q. Para ello, en primer lugar se calcula $\frac{\partial f_v}{\partial X_v}$, con lo que el jacobiano total queda como sigue:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{X}} = \begin{pmatrix} \frac{\partial f_v}{\partial X_v} & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \tag{6.48}$$

De forma similar se calcula la matriz de ruido de predicción total Q:

$$\mathbf{Q} = \begin{pmatrix} Q_v & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \tag{6.49}$$

Finalmente, una vez que se han calculado los términos de las ecuaciones 6.48 y 6.49 se puede calcular la predicción de la matriz de covarianza P como:

$$\hat{\mathbf{P}}\left(\mathbf{k}+\mathbf{1}|\mathbf{k}\right) = \left[\frac{\partial f}{\partial X} \cdot P(k|k) \cdot \left(\frac{\partial f}{\partial X}\right)^t\right] + Q \tag{6.50}$$

6.5. Modelo de Medida

El modelo de medida se basa en la obtención de las coordenadas 3D de posición de determinadas marcas naturales del entorno. La extracción de estas marcas se realiza utilizando uno de los operadores explicados en el capítulo 5. En la figura 6.12 se puede ver un esquema de las distintas fases que componen el modelo de medida.

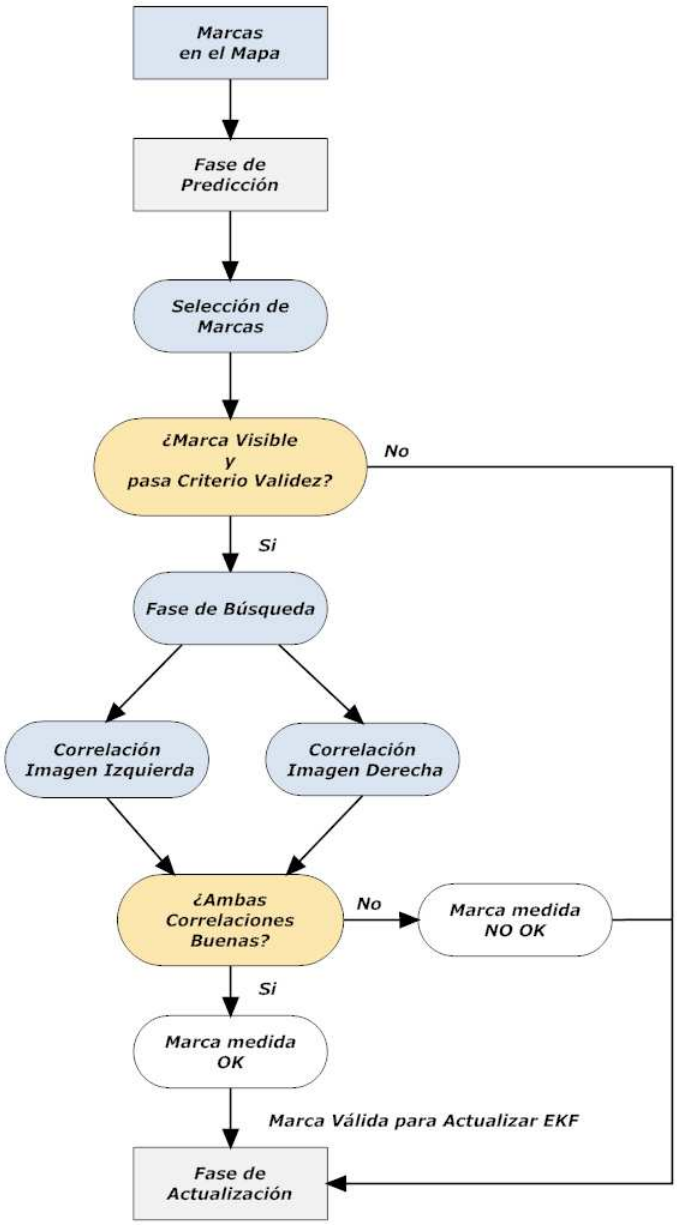


Figura 6.12: Esquema de las distintas fases del modelo de medida

6.5.1. Selección de Marcas

Antes de pasar a la medida de las marcas, es necesario realizar primeramente una fase de selección de marcas atendiendo principalmente a dos criterios: criterio de visibilidad y criterio de validez.

- **Criterio de Visibilidad**: De entre todas las marcas naturales capturadas previamente y que forman parte del mapa 3D del entorno, **se descartan aquellas marcas que no sean visibles**. Para pasar con éxito el criterio de visibilidad, la marca deberá cumplir con las siguientes tres condiciones:

  1. La proyección de la marca en la cámara izquierda (u_L, v_L) y en la cámara derecha (u_R, v_R) no deberá quedar fuera del **campo de visión**. Es decir, estas coordenadas deberán estar contenidas en su correspondiente cuadro de imagen menos un margen igual al tamaño del parche (ver figura 6.13).

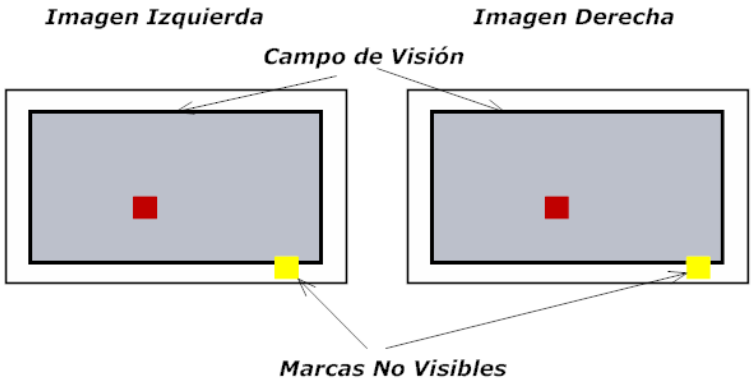


Figura 6.13: Campo de visión de las proyecciones de las marcas

  2. El **cambio de ángulo del punto de vista** no debe exceder un ángulo límite. Es decir, el ángulo formado por el vector de medida tomado cuando la marca fue inicializada y el vector de medida actual no debe exceder un valor determinado (normalmente 45º). Esta restricción es necesaria debido al cambio de la apariencia relativa al observar una imagen desde diferentes ángulos.

     El cálculo de dicho ángulo se realiza según la siguiente ecuación:

$$\beta = \cos^{-1}\left(\frac{h_i \cdot h_{iorig}}{|h_i|\,|h_{iorig}|}\right) \tag{6.51}$$

     En la ecuación 6.51, h_{iorig} es un vector que contiene las coordenadas 3D (x, y, z) de la marca con respecto a la posición de la cámara izquierda en el momento de su inicialización. Mientras, que h_i es un vector que contiene la predicción de las coordenadas 3D de la marca con respecto a la posición de la cámara izquierda en ese instante de tiempo.

  3. La **distancia de la cámara izquierda a la marca** puede ser superior o inferior a la distancia en el momento de su inicialización hasta un cierto límite, es decir, el cociente entre el módulo del vector de medida tomado cuando la marca fue inicializada y el módulo del vector de media actual debe estar entre un valor mínimo y máximo (normalmente entre 5/7 y 7/5). Esto es debido a que la apariencia de la marca es diferente según la distancia desde la que se observe. Dicho cociente se calcula como:

$$\mathbf{T} = \frac{|h_i|}{|h_{iorig}|} \tag{6.52}$$

     En la figura 6.14 se muestran gráficamente ambas condiciones.

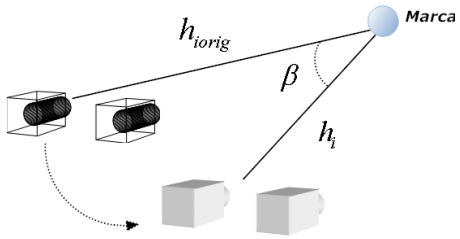


Figura 6.14: Condiciones de visibilidad de ángulo y módulo

- **Criterio de Validez**: Una vez escogidas aquellas marcas que pasan el criterio de visibilidad, el siguiente filtro consiste en **no seleccionar aquellas marcas cuyas medidas hayan sido fallidas más de un factor del** 70 % **de los intentos de medida realizados**. Se entiende por **marcas fallidas** aquellas cuya medida de correlación no resulta suficientemente buena. Las marcas que no superen esta condición serán eliminadas del mapa, ya que el objetivo es mantener a lo largo del tiempo, aquellas marcas más estables cuya apariencia sea relativamente constante de tal modo que se eviten posibles problemas de reflejos y oclusiones.

En la implementación, los parches correspondientes a cada marca llevan asociados un **código de colores** para identificar su propio estado. Este código es el siguiente:

- **Rojo**: Indica que la marca es visible, presenta una parametrización 3D y fue medida correctamente en la iteración anterior.
- **Naranja**: Indica que la marca es visible, presenta una parametrización inversa y fue medida correctamente en la iteración anterior.
- **Azul**: Indica que la marca es visible pero no fue correctamente medida en la iteración anterior.
- **Amarillo**: Indica que la marca no es visible.

6.5.2. Predicción

Para poder realizar la predicción de los vectores de medida de cada una de las marcas seleccionadas, es necesario conocer los siguientes valores:

- La posición absoluta en el sistema de referencia global de cada una de las marcas Y_i, en el caso de que las marcas presenten una parametrización 3D.
- La posición absoluta en el sistema de referencia global de la cámara en el momento que fue inicializada la marca X_{ori}, así como los valores de los parámetros $1/\rho$, y el vector unitario $m(\theta, \phi)$ para el caso en el que las marcas presenten una parametrización inversa.
- La posición absoluta de la cámara en el sistema de referencia global.

Si obtenemos la matriz de rotación R^{CW} a partir de los valores de orientación del vector X_p, se puede calcular la ecuación de predicción de la siguiente forma:

- Predicción para marcas con parametrización 3D:

$$\mathbf{h_i}_{[\mathbf{3,1}]} = R^{CW} \cdot (Y_i - X_{cam}) \tag{6.53}$$

- Predicción para marcas con parametrización inversa:

$$\mathbf{h_i}_{[\mathbf{3,1}]} = R^{CW} \cdot \left((X_{ori} - X_{cam}) + \frac{1}{\rho_i} \cdot m(\theta, \phi) \right) \tag{6.54}$$

6.5.3. Búsqueda

Una vez que se han obtenido las predicciones de la medidad de cada una de las marcas, el siguiente paso consiste en obtener la medidad real z_i. Para ello será necesario buscar la proyección de cada una de las marcas en un determinado área de búsqueda.

6.5.3.1. Obtención de Proyecciones y Jacobianos asociados

Como se ha visto en la sección 6.5.2, la predicción del valor de medida h_i viene definida por las coordenadas de cada marca (independientemente de si la parametrización es 3D o inversa) en el sistema de referencia de la cámara izquierda. En la figura 6.15 se puede observar el escenario en el cuál se encuentra una marca, las dos cámaras (izquierda y derecha) así como el sistema de referencia global y el local de cada una de las cámaras. La proyección de la marca en el plano imagen de la cámara izquierda (u_L, v_L) queda definida por la intersección del vector h_i con este mismo plano. Del mismo modo, se puede obtener la proyección de la marca en el plano imagen de la cámara derecha a partir del vector h_{iR}. Por lo tanto, para poder realizar las proyecciones es necesario acudir a las matrices de proyección de cada una de las cámaras que incluyen los parámetros intrínsecos de cada una de ellas:

$$\begin{pmatrix} su_L \\ sv_L \\ s \end{pmatrix} = \begin{pmatrix} FC1_L & 0 & CC1_L & 0 \\ 0 & FC2_L & CC2_L & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} h_{ix} \\ h_{iy} \\ h_{iz} \\ 1 \end{pmatrix} \tag{6.55}$$

$$\begin{pmatrix} su_R \\ sv_R \\ s \end{pmatrix} = \begin{pmatrix} FC1_R & 0 & CC1_R & 0 \\ 0 & FC2_R & CC2_R & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} h_{iRx} \\ h_{iRy} \\ h_{iRz} \\ 1 \end{pmatrix} \tag{6.56}$$

Un aspecto importante a tener en cuenta es el denominado *plano epipolar*, que queda delimitado por los vértices formados por la marca en cuestión y los centros de los dos sistemas de referencia de las cámaras. Dicho plano intersecta a los dos planos imagen en las denominadas *rectas epipolares*. La principal propiedad de estas rectas es que dada una marca en el espacio 3D, sus proyecciones en ambos planos imagen deberán pertenecer a dichas rectas. De tal modo, que dada una marca y su proyección en una de las cámaras, se debe realizar la búsqueda de su correspondiente proyección en la otra cámara sobre la recta epipolar de dicha cámara (ver 4.2.4).

Los pasos necesarios para obtener (u_L, v_L), (u_L, v_L), y sus jacobianos correspondientes $\frac{\partial U_L}{\partial h_i}$, $\frac{\partial U_R}{\partial h_i}$ son los siguientes:

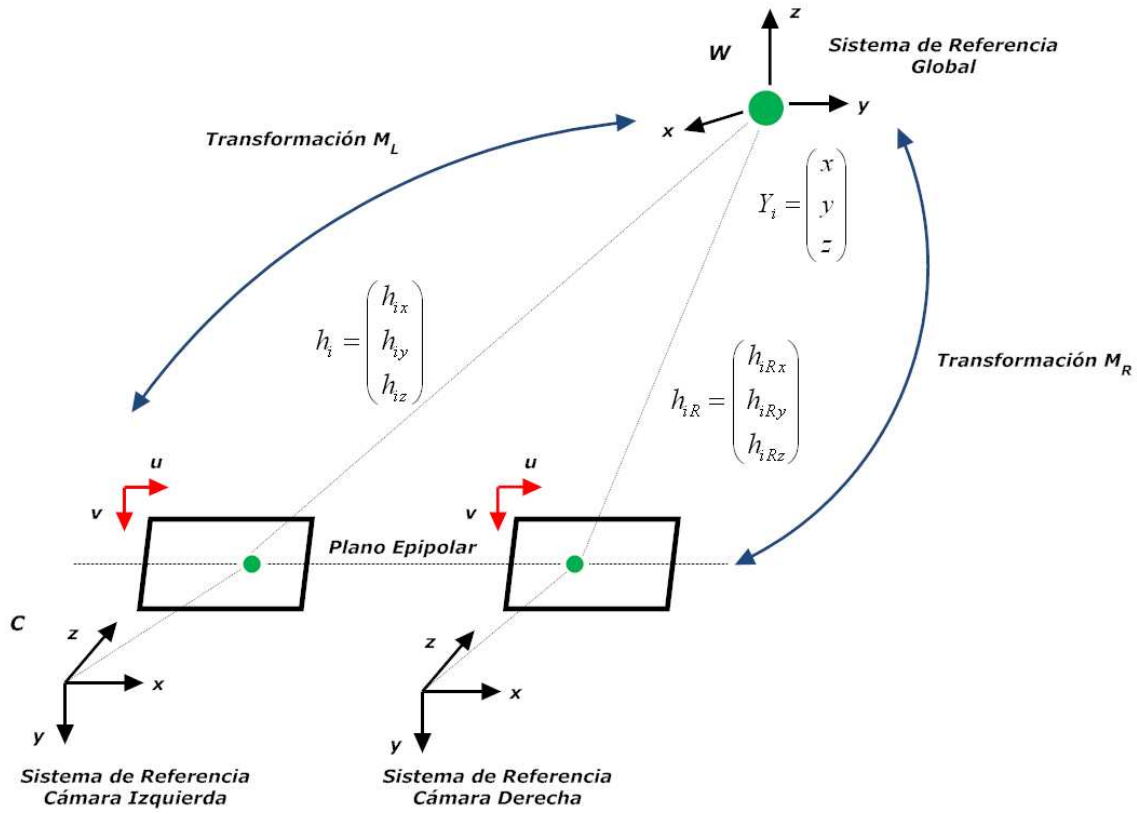


Figura 6.15: Representación de la geometría epipolar 3D y nomenclatura utilizada

- **Cámara Izquierda**:

  En primer lugar, es posible obtener las coordenadas de proyección directamente a partir de h_i a partir de la ecuación 6.55 obteniendo:

$$\begin{cases} \mathbf{u_L} &= FC1_L \cdot \frac{h_x}{h_z} + CC1_L \\ \mathbf{v_L} &= FC2_L \cdot \frac{h_y}{h_z} + CC2_L \end{cases} \tag{6.57}$$

  Para calcular los jacobianos, derivamos los términos correspondientes:

$$\frac{\partial \mathbf{U_L}}{\partial \mathbf{h_i}}_{[\mathbf{2,2}]} = \begin{pmatrix} u_L \\ v_L \end{pmatrix} = \begin{pmatrix} \frac{FC1_L}{h_z} & 0 & -\frac{FC1_L \cdot h_x}{h_z^2} \\ 0 & \frac{FC2_L}{h_z} & -\frac{FC2_L \cdot h_y}{h_z^2} \end{pmatrix} \tag{6.58}$$

- **Cámara Derecha**:

  Para obtener las coordenadas de imagen de la cámara derecha y su jacobiano asociado, dado que el sistemea de refencia de la cámara es coincidente con el sistema de referencia de la cámara izquierda, es necesario transformar h_i para poder expresarlo en el sistema de coordenadas de la cámara derecha. La relación entre las coordenadas de proyección de la cámara derecha y el vector de medida h_i teniendo en cuenta la matriz de rotación y el vector de traslación entre las cámaras izquierda y derecha es:

$$\begin{cases} \mathbf{u_R} &= FC1_R \cdot \frac{R_{11}h_x+R_{12}h_y+R_{13}h_z+T_x}{R_{31}h_x+R_{32}h_y+R_{33}h_z+T_z} + CC1_R \\ \mathbf{v_R} &= FC2_R \cdot \frac{R_{21}h_x+R_{22}h_y+R_{23}h_z+T_y}{R_{31}h_x+R_{32}h_y+R_{33}h_z+T_z} + CC2_R \end{cases} \tag{6.59}$$

Para calcular el jacobiano $\frac{\partial U_R}{\partial h_i}$ siendo $U_R = (u_R\ v_R)^t$ simplemente hay que derivar con respecto a los términos correspondientes. [1]

6.5.3.2. Obtención del Área de Búsqueda

Una vez que se han obtenido las proyecciones en el plano imagen de ambas cámaras, resultado de la fase de predicción, el siguiente paso a realizar es definir el área de búsqueda de la marca centrada en la predicción de las coordenadas de imagen.

Para calcular la incertidumbre en la predicción de las coordenadas de imagen en cada cámara, es necesario relacionarla con la incertidumbre en la medida de la posición de la marca en cuestión. Esta incertidumbre es denominada como **covarianza de innovación** S_i, la cuál proviene básicamente de tres fuentes distintas:

- Incertidumbre en la posición real de la cámara izquierda P_{XX}.
- Incertidumbre en la posición real de la marca $P_{Y_iY_i}$.
- Incertidumbre en la medida de dicha posición (ruido de medida) R_i.

Si tenemos en cuenta las covarianzas cruzadas, el cálculo de la covarianza de innovación queda como sigue (ver referencias [4]):

$$\begin{aligned} \mathbf{S_{i\ [3,3]}} = & \frac{\partial h_i}{\partial X_v} \cdot P_{XX} \cdot \left(\frac{\partial h_i}{\partial X_v}\right)^t + \frac{\partial h_i}{\partial X_v} \cdot P_{XY_i} \cdot \left(\frac{\partial h_i}{\partial Y_i}\right)^t \\ & \frac{\partial h_i}{\partial Y_i} \cdot P_{Y_iX} \cdot \left(\frac{\partial h_i}{\partial X_v}\right)^t + \frac{\partial h_i}{\partial Y_i} \cdot P_{Y_iY_i} \cdot \left(\frac{\partial h_i}{\partial Y_i}\right)^t + R_i \end{aligned} \tag{6.60}$$

El siguiente paso consiste en transformar la incertidumbre en la medida, calculada anteriormente, en la incertidumbre en su proyección en ambas cámaras U_L y U_R. Para ello solamente es necesario realizar la siguiente transformación de la covarianza de innovación de la marca utilizando los jacobianos calculados anteriormente:

$$\mathbf{P_{U_L\ [2,2]}} = \frac{\partial U_L}{\partial h_i} \cdot S_i \cdot \left(\frac{\partial U_L}{\partial h_i}\right)^t \tag{6.61}$$

$$\mathbf{P_{U_R\ [2,2]}} = \frac{\partial U_R}{\partial h_i} \cdot S_i \cdot \left(\frac{\partial U_R}{\partial h_i}\right)^t \tag{6.62}$$

Las dos covarianzas anteriormente calculadas definen sendas densidades de probabilidad Gaussianas de media U_L y U_R respectivamente.

Si se restringe el área de búsqueda a un número de desviaciones típicas, se obtiene de forma general, una elipse para cada una de las proyecciones, cuya ecuación de puede obtener de la siguiente forma:

[1] Debido a que los términos del jacobiano resultante son expresiones muy largas, no se incluyen los cálculos en el presente trabajo.

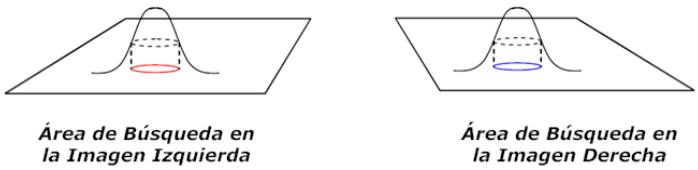


Figura 6.16: Áreas de búsqueda Gaussianas

1. Partiendo de la ecuación de la Normal de dos variables y de media 0, por ejemplo para la cámara izquierda:

$$\mathbf{P_r}(\mathbf{U}) = \frac{1}{\sqrt{(2\pi)^2 |P_{U_L}|}} \cdot e^{-\frac{1}{2}\left(U_L^t \cdot P_{U_L}^{-1} \cdot U_L\right)} \tag{6.63}$$

2. Si se toma la restricción de búsqueda a $n \cdot \sigma$, es decir, aquellos puntos que cumplen una *distancia de Mahalanobis* al origen menor que n, se puede obtener la siguiente expresión:

$$U_L^t \cdot P_{U_L}^{-1} \cdot U_L \leq n^2 \tag{6.64}$$

3. Desarrollando la ecuación anterior, se obtiene la expresión que delimita el área de búsqueda pixélica (u, v) donde realizar la búsqueda:

$$u^2 \cdot P_{U_{L11}}^{-1} + 2 \cdot u \cdot v \cdot P_{U_{L12}}^{-1} + v^2 \cdot P_{U_{L22}}^{-1} < n^2 \tag{6.65}$$

4. La mitad del ancho y el alto de las elipses se pueden calcular a partir de las siguientes ecuaciones:

$$\mathbf{Width_{Half}} = \frac{n \cdot \sigma}{\sqrt{P_{U_{L11}}^{-1} - \frac{\left(P_{U_{L11}}^{-1}\right)^2}{P_{U_{L22}}^{-1}}}} \tag{6.66}$$

$$\mathbf{Height_{Half}} = \frac{n \cdot \sigma}{\sqrt{P_{U_{L22}}^{-1} - \frac{\left(P_{U_{L12}}^{-1}\right)^2}{P_{U_{L11}}^{-1}}}} \tag{6.67}$$

A la hora de realizar la búsqueda, será necesario desplazar las elipses calculadas hasta el valor de la media U_L, U_R según corresponda. El número de desviaciones típicas n que se considera para el área de búsqueda, en este trabajo se ha fijado a un valor de 5. Es necesario llegar a un compromiso entre ruido de incertidumbre, y área de búsqueda dependiendo del sistema a modelar. Sistemas más sencillos de modelar y con menos grados de libertad pueden permitir un valor menor de este parámetro n.

6.5.3.3. Método de Correlación

El objetivo de definir unas áreas de búsqueda de máxima probabilidad, es encontrar en estas áreas de búsqueda el parche que presente una mejor correlación con el parche extraído de la

imagen original. Para ello se parte de la imagen de un parche de tamaño $B \times B$ y la región de búsqueda calculada en la nueva imagen y centrada en los valores de las proyecciones obtenidas tras la fase de predicción. Antes de realizar la correlación, será necesario modificar la apariencia del parche con el nuevo punto de vista, como se explica en la sección 6.6.

Para cada uno de los puntos de la región de búsqueda, se realiza una correlación *ZMNCC* (ver sección 4.2.8). La correlación *ZMNCC* puede tomar valores entre 0 (mínima correlación) y 1 (máxima correlación), de tal modo que si el valor máximo de correlación para alguno de los puntos del área de búsqueda supera un determinado umbral (por ejemplo 0,85 la correlación para esa proyección es dada por buena.

En la figura 6.17 se muestra el proceso de correlación píxel a píxel, en donde (i_{offset}, j_{offset}) indica la posición del parche a evaluar respecto a la posición del parche original a buscar (i_0, j_0). Teniendo en cuenta que el parche es siempre cuadrado: $B = i_{max} - i_0 = j_{max} - j_0$.

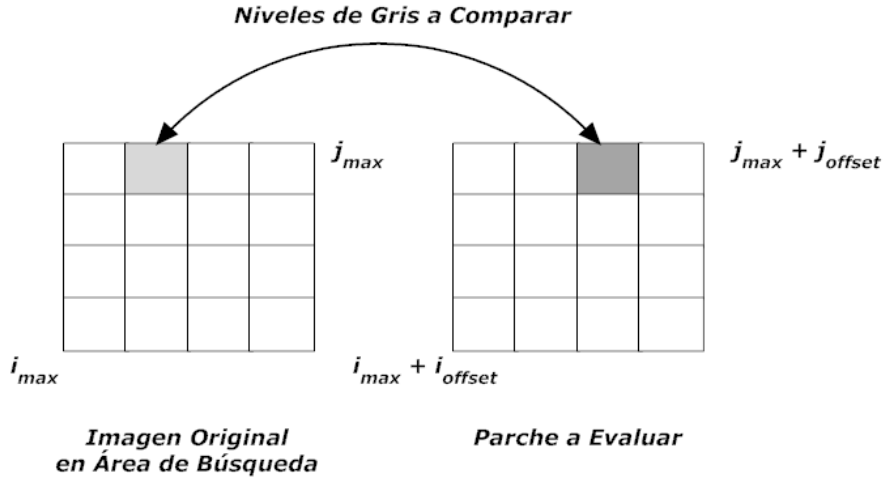


Figura 6.17: Proceso de correlación píxel a píxel

Dependiento de los distintos métodos de adaptación de parches, se pueden llevar a cabo las siguientes búsquedas:

- Si **no se realiza adaptación de parches**, es necesario realizar la correlación del parche original izquierdo en el área de búsqueda izquiero y del parche original derecho en el área de búsqueda derecha. Para que una marca sea medida correctamente, es necesario que el valor de la correlación en ambas zonas de búsqueda supere el umbral de correlación.

- Si se aplica el método del **parche adaptado**, será necesario modificar la apariencia tanto del parche original izquierdo como del derecho, y realizar sus respectivas correlaciones en sus respecitvas regiones de búsqueda. Del mismo modo, para que una marca sea medida correctamente, es necesario que el valor de ambas correlaciones supere el umbral de correlación.

- Si se utiliza el método de **warping mediante homografía** únicamente es necesario adaptar el parche original izquierdo y buscar el valor de correlación máximo en la región de búsqueda izquierda. Si el valor de correlación es bueno, a continuación, se busca la correspondencia del punto encontrado en la imagen derecha mediante una búsqueda sobre la recta epipolar derecha dentro de la región de búsqueda derecha.

6.5.4. Actualización

Una vez obtenido el vector de medida z_i, antes de realizar el proceso de actualización, es necesario construir los vectores y matrices implicados:

1. Si se define el vector $\eta = z_i - h_i$ como la diferencia entre el vector de medida real z_i y el vector de predicción h_i, es posible construir el vector η_{tot} correspondiente al conjunto de todas las **marcas medidas correctamente** de la siguiente forma:

$$\eta_{\mathbf{tot}\ [\mathbf{n_{tot}}\cdot\mathbf{3},\mathbf{1}]} = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \cdots \\ \eta_{n_{tot}} \end{pmatrix} \tag{6.68}$$

   en donde n_{tot} es el número de marcas (tanto con parametrización 3D como inversa) que han sido correctamente medidas.

2. El segundo elemento necesario para la actualización del filtro es el jacobiano $\frac{\partial h}{\partial X_{tot}}$. Su construcción se realiza de la siguiente manera:

$$\frac{\partial \mathbf{h}}{\partial \mathbf{X_{tot}}} = \begin{pmatrix} \frac{\partial h_1}{\partial X_v} & \frac{\partial h_1}{\partial Y_1} & 0 & \cdots & 0 \\ \frac{\partial h_2}{\partial X_v} & 0 & \frac{\partial h_1}{\partial Y_2} & \cdots & 0 \\ \vdots & 0 & 0 & \ddots & 0 \\ \frac{\partial h_{n_{tot}}}{\partial X_v} & 0 & 0 & \cdots & \frac{\partial h_{n_{tot}}}{\partial Y_{n_{tot}}} \end{pmatrix} \tag{6.69}$$

3. Por último, es necesario calcular la covarianza de innovación total S, para lo cuál es necesario antes calcular la matriz de ruido de medida total R_{tot}:

$$\mathbf{R_{tot}} = \begin{pmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & R_{n_{tot}} \end{pmatrix} \tag{6.70}$$

   Para calcular S, simplemente hay que transformar la matriz de covarianza total y sumarle el ruido de medida total:

$$\mathbf{S} = \left[\frac{\partial h}{\partial X_{tot}} \cdot P \cdot \left(\frac{\partial h}{\partial X_{tot}} \right)^t \right] + R_{tot} \tag{6.71}$$

Una vez construidas las matrices necesarias para la actualización, sólo es necesario aplicar la fórmula de actualización del EKF, tal y como se describe en [36]:

$$\begin{cases} \widehat{\mathbf{X}}_{\mathbf{new}} = \widehat{X}_{old} + W \cdot \eta_{tot} \\ \mathbf{P_{new}} = P_{old} - W \cdot S \cdot W^t \end{cases} \tag{6.72}$$

en donde la matriz W tiene la siguiente expresión:

$$\mathbf{W} = P \cdot \left(\frac{\partial h}{\partial X_{tot}} \right)^t \cdot S^{-1} \tag{6.73}$$

6.6. Transformación de la Apariencia del Parche

Uno de los puntos claves para un buen funcionamiento de un algoritmo de SLAM basado en información visual, es que las marcas que forman parte del mapa 3D sean marcas de buena calidad, fáciles de ser identificadas a lo largo del tiempo.

En el presente trabajo, el *matching* de las marcas se basa en una simple correlación de parches 2D. Si no se tiene en cuenta el cambio de la apariencia del parche a medida que el punto de vista cambia, esto implica que el número de movimientos de la cámara y los distintos puntos de vista posibles sean muy limitados, lo que implicaría a su vez añadir nuevas marcas al mapa incrementando el coste computacional y la incertidumbre del mapa.

Por lo tanto, es necesario realizar una transformación de la apariencia del parche en cada iteración con el fin de poder seguir a una marca durante un mayor número de frames. Para ello en el presente trabajo se estudian dos métodos denominados: Parche Adaptado y Warping mediante Homografía. A continuación se explican las principales características de cada uno de ellos.

6.6.1. Parche Adaptado

En el momento de inicializar una marca, se estiman las posiciones 3D respecto del sistema de referencia global de cada uno de los píxeles del parche, bajo la suposición de que el parche es localmente plano. Posteriormente, en siguientes iteraciones cuando cambia el punto de vista de la cámara, se estiman las proyecciones de cada uno de los píxels del parche, teniendo en cuenta el estado actual de la cámara izquierda y las posiciones 3D almacenadas de cada uno de los píxeles del parche.

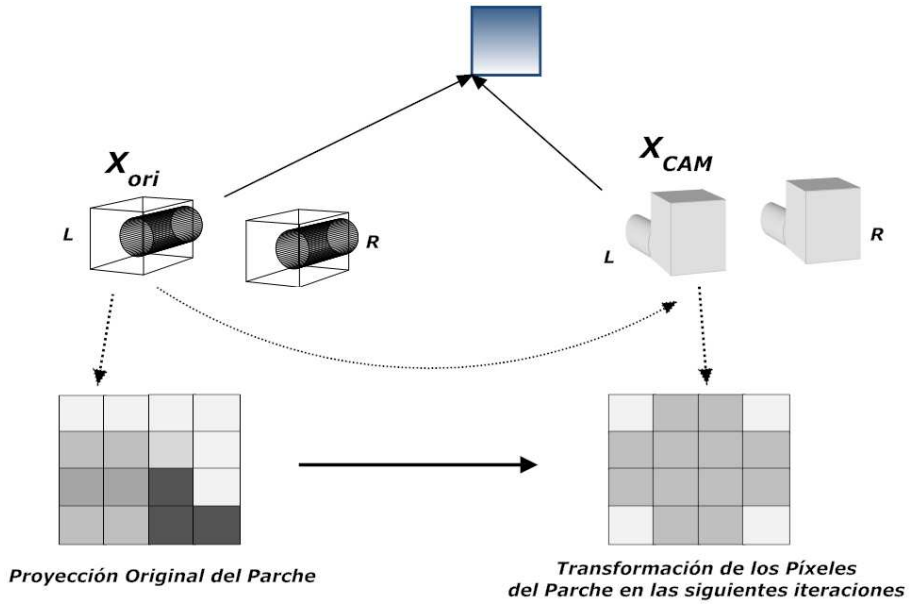


Figura 6.18: Transformación de la apariencia del parche

Una vez calculadas las nuevas proyecciones de los píxels del parche, antes de realizar la búsqueda por correlación, es necesario asignar un valor a todos los píxels que quedaron sin correspondencia en el paso anterior. Para ello se utiliza un método de interpolación por vecindad. Dicho método se explica a continuación:

1. En primer lugar, se identifican todos los píxels vacíos. Es decir, aquellos píxels dentro del parche para los cuáles no se ha obtenido una proyección.

2. Posteriormente, para cada uno de dichos píxels, se realiza una búsqueda progresiva del píxel ocupado más cercano, tal y como se muestra en la figura 6.19.

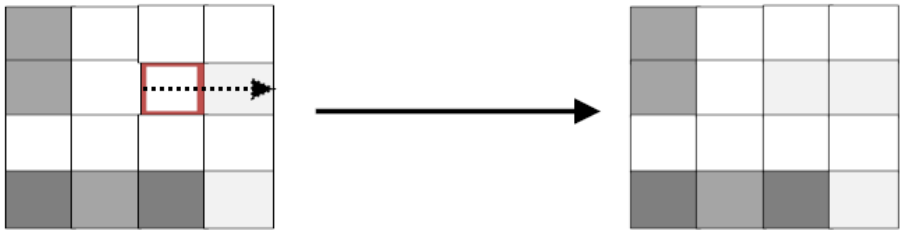


Figura 6.19: Interpolación del parche por vecindad

3. Finalmente, el píxel vacío tomará el valor del píxel ocupado anteriormente.

6.6.2. Warping mediante Homografía

Cada uno de los parches que han sido seleccionados como marcas naturales se corresponden con una observación de una superficie localmente plana en el entorno 3D, en vez de considerar únicamente una imagen 2D. Esta aproximación de superficie plana es relativa al tipo de movimiento de la cámara sobre la marca observada, pero sin embargo, muchas marcas en diversos entornos cumplen esta característica bajo tipos de movimiento normales.

Cuando es necesario realizar una medida de una marca, se puede utilizar para esta medida la estimación actual de la posición y orientación de la cámara izquierda que se obtiene a partir del vector de estado del proceso de SLAM. A partir de estos datos, y de la normal a la superficie del plano que contiene a la marca en el entorno 3D, se puede modificar la apariencia del parche mediante un *warping*.

El primer paso para poder realizar el *warping* es calcular el vector normal a la superficie que contiene a la marca 3D. El método desarrollado en el presente trabajo, se basa en los métodos utilizados en [37], [38]. En [37] se propone un método basado en un sistema monocular, en el cuál se calcula la normal a la superficie entre dos vistas mediante el uso de alineamiento de imágenes utilizando métodos de descenso de gradiente, mientras que en [38] se propone un método de cálculo de homografías planas mediante visión para ayuda a un sistema de navegación de un robot móvil.

En la figura 6.20 se expone el planteamiento del problema del cálculo de la normal a la superficie de un plano que contiene a un punto 3D, y del cálculo de las homografías correspondientes entre las cámaras izquierda y derecha para un mismo instante de tiempo, así como la homografía existente entre distintas vistas para la cámara izquierda.

Supongamos que una de las marcas detectadas por el sistema se encuentra en un plano π que no pasa por el origen. Por lo tanto, la ecuación de este plano vendrá definida por:

$$\pi : a \cdot x_1 + b \cdot y_1 + c \cdot z_1 + 1 = 0 \tag{6.74}$$

La transformación entre dos sistemas de coordenadas diferentes, se puede modelar mediante una matriz de rotación y un vector de traslación. Por ejemplo, supongamos la transformación

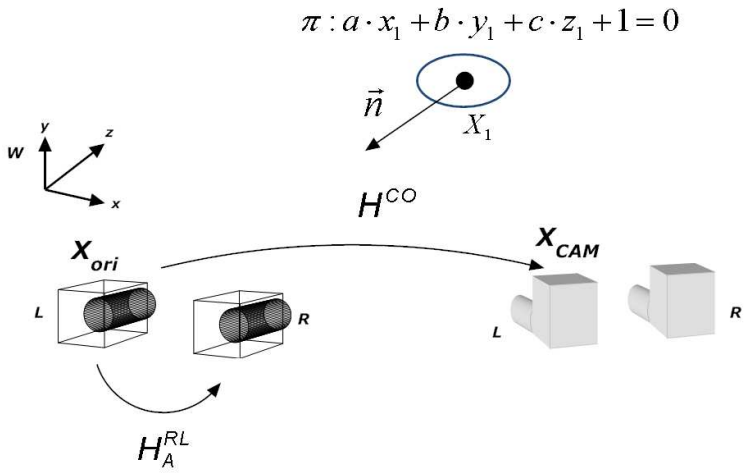


Figura 6.20: Geometría del par estéreo y superficies localmente planas

existente entre dos sistemas de coordenadas genéricos:

$$X_2 \;=\; R \cdot X_1 + T \tag{6.75}$$

Si el punto X_1 pertenece al plano π obtendremos la relación siguiente:

$$n^t \cdot X_1 \;=\; -1 \tag{6.76}$$

Si sustituimos la expresión anterior en la ecuación 6.75 podemos obtener la homografía existente entre los dos sistemas de coordenadas X_1 y X_2.

$$X_2 \;=\; R \cdot X_1 + T \;=\; R \cdot X_1 - T \cdot n^t \cdot X_1 \;=\; \left(R - T \cdot n^t\right) \cdot X_1 \tag{6.77}$$

Para obtener una transformación en base a las coordenadas pixélicas de proyección, simplemente tenemos que multiplicar por las respectivas matrices de proyección en el orden adecuado:

$$U_2 \;=\; C_2 \cdot \left(R - T \cdot n^t\right) \cdot C_1^{-1} \cdot U_1 \tag{6.78}$$

Una vez explicado el método entre dos sistemas de coordenadas genéricos, a continuación se explica el método desarrollado:

1. La relación existente entre puntos de la cámara izquierda y puntos de la cámara derecha viende dada por la siguiente expresión:

$$U_R \;=\; C_R \cdot \left(R^{RL} - T^{RL} \cdot n^t\right) \cdot C_L^{-1} \cdot U_L \tag{6.79}$$

   La ecuación anterior depende de la matriz de rotación R^{RL} y el vector de traslación T^{RL}. Estos parámetros son conocidos y además el error existente en la determinación de estos parámetros es muy bajo, ya que se han obtenido previamente en el proceso de calibración del par estéreo.

2. Si suponemos que existe una transformación afín entre el parche visto por la cámara izquierda y la cámara derecha podemos expresar la transformación afín H_A^{RL} como:

$$\mathbf{H_A^{RL}} \;=\; C_R \cdot \left(R^{RL} - T^{RL} \cdot n^t\right) \cdot C_L^{-1} \tag{6.80}$$

3. Se puede calcular la transformación afín entre ambas cámaras a partir de 3 correspondencias de puntos no conlineales. Suponiendo los parches localmente planos, y a partir de las correspondencias entre 3 puntos se obtiene la transformación afín entre cámaras H_A^{RL}. Como se puede observar en la ecuación 6.80, en el cálculo de esta transformación está implícito el cálculo del vector normal n del plano π.

4. A partir de la ecuación 6.80 podemos despejar el valor del producto de los vectores $T^{RL} \cdot n^t$. Llamando a esta matriz X, podemos calcularla a partir de la siguiente expresión:

$$\mathbf{X} \;=\; T^{RL} \cdot n^t \;=\; R^{RL} - C_R^{-1} \cdot H_A^{RL} \cdot C_L \tag{6.81}$$

5. En la ecuación anterior conocemos todos los parámetros implicados, ya que el valor de la transformación afín H_A^{RL} ha sido calculado previamente, y el resto de matrices implicadas se conocen del proceso de calibración estéreo, se obtiene un sistema de 9 ecuaciones con 3 incógnitas que son las componentes del vector normal al plano π.

$$\begin{cases} n_x = \frac{X_{11}}{T_x} & n_x = \frac{X_{21}}{T_y} & n_x = \frac{X_{31}}{T_z} \\ n_y = \frac{X_{12}}{T_x} & n_y = \frac{X_{22}}{T_y} & n_y = \frac{X_{32}}{T_z} \\ n_z = \frac{X_{13}}{T_x} & n_z = \frac{X_{23}}{T_y} & n_z = \frac{X_{33}}{T_z} \end{cases} \tag{6.82}$$

6. En el momento de inicializar una marca, se calcula también el vector normal del plano π que contiene a la marca. Una vez que se conoce el vector normal n se puede calcular la homografía existente entre dos puntos de vista distintos tomando como referencia la cámara izquierda a partir de la siguiente ecuación:

$$U_{CAM} \;=\; C_L \cdot \left(R^{CO} - T^{CO} \cdot n^t\right) \cdot C_L^{-1} \cdot U_{ORI} \tag{6.83}$$

en donde la matriz de rotación R^{CO} y el vector de traslación T^{CO} son conocidos ya que pueden ser calculados a través de la información del vector de estado que nos proporciona el EKF.

6.7. Inicialización de Nuevas Marcas

Una parte importante en la implementación del sistema es la captura de nuevas marcas. Dado que el sistema **no parte de ninguna marca conocida a priori** inicialmente el vector de estado total estará compuesto únicamente por el vector de estado de la cámara izquierda. Inicialmente se supone que la cámara izquierda se encuentra en el origen de coordenadas del sistema global, con una orientación por defecto (línea visual paralela al eje Z). En cuanto a la **velocidad lineal y angular**, **es necesario partir de un valor inicial no nulo** para permitir la convergencia del filtro. La matriz de covarianza total P se supone inicialmente nula, ya que se parte de una posición inicial preestablecida.

El primer paso necesario para comenzar la ejecución del filtro, consiste en añadir la primera marca al filtro. A la hora de de buscar el parche correspondiente a la mejor marca para añadir al mapa, la región de búsqueda a tener en cuenta será toda la imagen correspondiente a la cámara izquierda menos un margen en los bordes exteriores del tamaño del propio parche B. En este trabajo, el tamño de los parches será de 11×11 píxels (ver figura 6.21). Para calcular el valor de las matrices de covarianza asociadas a esta marca es necesario seguir los pasos que se explican en la sección 6.9.

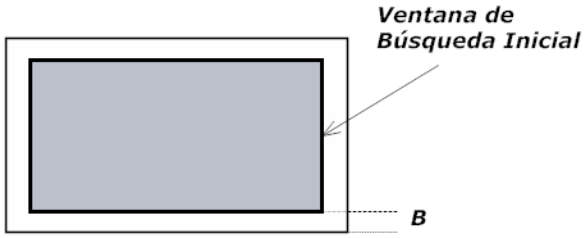


Figura 6.21: Ventana de búsqueda inicial para selección de nuevas marcas

A la hora de introducir los elementos de la nueva marca en el proceso del filtro, el procedimiento a seguir se explica en la sección 6.10.

Una vez capturada e inicializada la primera marca, en cada iteración del filtro se irán capturando nuevas marcas en función de los siguientes parámetros:

- **Número de marcas visibles**: El mínimo número de marcas visibles no debe ser inferior a un valor predeterminado, en este caso 10. Por lo tanto, siempre que existan menos de 10 marcas visibles en una iteración del filtro, será necesario incorporar una nueva marca en el proceso.
- **Número de marcas correctamente medidas**: Tal y como se explica en la sección 6.5.3.3, en cada iteración del filtro se intenta medir la posición de las marcas visibles a través de la correlación en el área de búsqueda de cada marca. Si la medida es fallida, dicha marca no es utilizada a la hora de actualizar el filtro. Si el número de marcas exitosamente medidas es inferior a un cierto valor, en este caso 10, es posible perder la convergencia del filtro después de varias iteraciones. En este caso, por lo tanto también se procede a buscar una nueva marca para añadir al sistema.

Llegado el momento de **buscar una nueva marca**, el proceso es el siguiente:

1. Se genera una región de búsqueda rectangular de dimensiones 50×50 situada aleatoria-

mente con distribución uniforme dentro de la imagen, excluyendo los márgenes anteriormente descritos (ver figura 6.22).

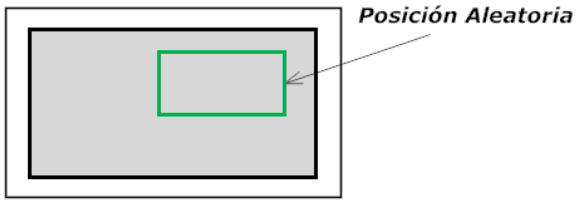


Figura 6.22: Región de búsqueda rectangular aleatoria

2. Posteriormente, se comprueba si existe alguna marca visible y medida correctamente dentro de la región de búsqueda. Si es así, se descarta dicha región y se vuelve a generar otra región aleatoriamente. Esto es así, para evitar la acumulación de marcas en un área visual pequeña, ya que la incertidumbre en la posición global de la cámara tiende a ser mayor en este caso (ver figura 6.23).

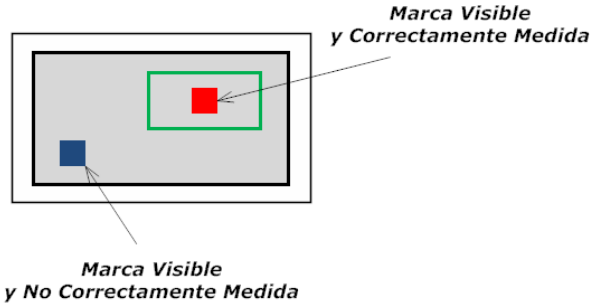


Figura 6.23: Comprobación de marcas en la región de búsqueda

3. En caso contrario, se procede a la búsqueda de la marca. Si el parche que se encuentra en esta región no es lo suficientemente bueno, también se descarta esta región y se vuelve a generar otra región aleatoriamente.

4. En los dos casos anteriores, se realizarán hasta un máximo de 15 intentos de obtención de nueva marca. Si se sobrepasa dicho límite, se permite avanzar al filtro una iteración más para volver a intentar añadir una nueva marca. En caso contrario, se almacenan las coordenadas de imagen de la nueva marca capturada y se procede a obtener su vector de estado y covarianza tal y como se explica en las secciones 6.8 y 6.9.

Además, será necesario almacenar en el proceso de captura de una nueva marca las imágenes asociadas a la marca (parche izquierdo y parche derecho) para poder realizar las correlaciones necesarias en el proceso de búsqueda. Es importante destacar, que los parches permanecen constantes durante toda la vida de la marca, es decir, no se actualizan con las nuevas vistas. Esto es así, con el fin de evitar el error acumulativo que conllevaría dicha actualización a la hora de localizar la marca.

Interesa buscar marcas cuyas características sean lo mejor posibles, para poder realizarlas un seguimiento durante el mayor número de frames posible. A la hora de buscar una nueva marca, se utiliza uno de los métodos de detección de características explicados en el capítulo 5.

6.8. Obtención del Vector de Estado de la Marca

En esta sección, se exponen los pasos necesarios para obtener los valores del vector de estado de una marca, considerando los dos posibles tipos de parametrizaciones: parametrización 3D y parametrización inversa.

6.8.1. Marcas con Parametrización 3D

Para obtener las coordenadas $Y_{i\ 3D}$ (vector de estado) de una marca con parametrización 3D, se siguen los siguientes pasos:

$$\mathbf{Y_{i\ 3D\ [3,1]}} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{6.84}$$

6.8.1.1. Búsqueda de la Correspondencia Epipolar

Una vez obtenidas las coordenadas de la marca en la cámara izquierda (u_L, v_L), es necesario calcular las coordenadas (u_R, v_R) de la marca correspondiente en la cámara derecha. Para encontrar dichas coordenadas en la imagen derecha, se debe buscar a lo largo de la *recta epipolar* derecha (ver sección 4.2.6.

1. El primer paso es obtener la ecuación de la recta epipolar. Para ello se define la ecuación de la siguiente manera:

$$a \cdot x + b \cdot y + c \cdot z = 0 \tag{6.85}$$

Para calcular los tres coeficientes, basta con multiplicar las coordenadas (u_L, v_L) de la cámara izquierda por la matriz fundamental F:

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = F \cdot \begin{pmatrix} u_L \\ v_L \\ 1 \end{pmatrix} \tag{6.86}$$

Para calcular la matriz fundamental F se utiliza la siguiente expresión en la cuál es necesario conocer el valor de la matriz esencial E y de las matrices de proyección de cada una de las cámaras:

$$F = \left(C_R^{-1}\right)^t \cdot E \cdot C_L^{-1} \tag{6.87}$$

en donde si recordamos las expresiones de las matrices implicadas:

$$C_L = \begin{pmatrix} FC1_L & 0 & CC1_L \\ 0 & FC2_L & CC2_L \\ 0 & 0 & 1 \end{pmatrix} \tag{6.88}$$

$$C_R = \begin{pmatrix} FC1_R & 0 & CC1_R \\ 0 & FC2_R & CC2_R \\ 0 & 0 & 1 \end{pmatrix} \tag{6.89}$$

$$E = \begin{pmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{pmatrix} \cdot R^{RL} \tag{6.90}$$

2. Una vez calculada la ecuación de la recta epipolar, el siguiente paso es la búsqueda de la correspondencia a lo largo de dicha recta. Es decir, la correlación de la imagen asociada a la marca tomada por la cámara izquierda (parche izquierdo), con imágenes de parches de las mismas dimensiones situadas entorno a la recta epipolar en la imgen tomada por la cámara derecha.

3. Una vez hallado el punto que presenta una correlación máxima, se comprueba si este valor supera un cierto umbral. Si es así, se obtienen las coordenadas (u_R, v_R) correspondientes para poder calcular la posición 3D de la marca.

6.8.1.2. Obtención de la Posición Absoluta Y_i de la Marca

Tal y como se explica en la sección 4.2.3 podemos obtener la posición 3D relativa al sistema de coordenadas de la cámara izquierda h_i a partir de las proyecciones pixélicas en la cámara izquierda (u_L, v_L) y en la cámara derecha (u_R, v_R).

$$\mathbf{h_{i\ [3,1]}} = \left(A^t A\right)^{-1} \cdot A^t b \tag{6.91}$$

Una vez obtenido el vector h_i, podemos obtener la posición absoluta Y_i de la marca en el sistema global de acuerdo con la transformación existente entre ambos sistemas de coordenadas:

$$\mathbf{Y_{i\ [3,1]}} = R^{WC} \cdot h_i + X_{cam} \tag{6.92}$$

Cada vez que se quiera obtener el vector para una marca con parametrización 3D, será necesario recalcular la matriz R^{WC} a partir de la información de orientación y obtener la posición de la cámara izquierda X_{cam} a partir de la información del vector de estado. La relación existente entre los cuaterniones y su correspondiente matriz de rotación, se puede consultar en el apéndice A.

6.8.2. Marcas con Parametrización Inversa

El cálculo a seguir para obtener el valor del vector estado de una marca con parametrización inversa es el siguiente:

$$\mathbf{Y_{i\ INV\ [6,1]}} = \begin{pmatrix} X_{ori} \\ \theta \\ \phi \\ 1\ /\ \rho \end{pmatrix} \tag{6.93}$$

- $\mathbf{X_{ori}}$: Es la posición 3D de la cámara en el sistema de referencia global, en el momento en el que la marca es inicializada.

- $\theta_\mathbf{i}$: El ángulo de azimuth se calcula a partir de la posición 3D de la cámara y de la marca en el sistema de referencia global en el momento en el que la marca es inicializada. Sea h^W

un vector 3D obtenido tras realizar la diferencia entre la posición 3D de la marca y de la cámara en el sistema de referencia global, la expresión para calcular el ángulo de azimuth es:

$$\theta_{\mathbf{i}} \;=\; tan^{-1}\left(\frac{h_z^W}{h_x^W}\right) \tag{6.94}$$

- $\phi_{\mathbf{i}}$: Del mismo modo podemos calcular el ángulo de elevación a partir de la siguiente expresión:

$$\phi_{\mathbf{i}} \;=\; -tan^{-1}\left(\frac{\sqrt{h_x^{W2}+h_z^{W2}}}{h_y^2}\right) \tag{6.95}$$

- $\mathbf{1}/\rho_{\mathbf{i}}$: El ángulo de azimuth se calcula a partir de la posición 3D de la cámara y de la marca en el sistema de referencia global en el momento en el que la marca es inicializada.

$$\frac{1}{\rho_i} = d_i \;=\; h_z^W \tag{6.96}$$

6.9. Obtención de la Covarianza del Vector de Estado de la Marca

A la hora de capturar una nueva marca e incluirla en el Filtro de Kalman, se necesita calcular tres tipos de covarianzas parciales:

- La covarianza del vector de estado de la nueva marca Y_i con el vector de estado de la cámara completo X_v (ver sección 6.1).
- Las covarianzas cruzadas de la nueva marca Y_i con el resto de marcas ya almacenadas Y_j, es decir, $P_{Y_jY_i}$ y $P_{Y_iY_j}$.
- La covarianza de la propia marca P_{YY}.

Los dos primeros tipos de covarianza serán estudiados en la sección 6.10, mientras que en la presente sección se estudia la obtención de P_{YY}. Del mismo modo, también se consideran las dos posibles parametrizaciones de las marcas.

6.9.1. Marcas con Parametrización 3D

La autocovarianza de Y_i esta compuesta por dos componentes:

1. En primer lugar, la debida a la **incertidumbre en el estado del robot**, es decir la autocovarianza P_{XX}. Para su cálculo es necesario transformar dicha incertidumbre en la correspondiente a la determinación de la situación de la nueva marca Y_i. Es decir:

$$\frac{\partial Y_i}{\partial X_v} P_{XX} \left(\frac{\partial Y_i}{\partial X_v}\right)^t \tag{6.97}$$

2. En segundo lugar, una **componente de ruido aleatorio** debida a la incertidumbre en la propia medida realizada por las cámaras.

$$\frac{\partial Y_i}{\partial h_i} R_i \left(\frac{\partial Y_i}{\partial h_i}\right)^t \tag{6.98}$$

Finalmente obtenemos la expresión de la autocovarianza como sumna de los dos términos anteriores:

$$\mathbf{P_{YY}}_{[\mathbf{3,3}]} = \frac{\partial Y_i}{\partial X_v} P_{XX} \left(\frac{\partial Y_i}{\partial X_v}\right)^t + \frac{\partial Y_i}{\partial h_i} R_i \left(\frac{\partial Y_i}{\partial h_i}\right)^t \tag{6.99}$$

Por lo tanto, a la hora de inicializar una nueva marca es necesario calcular $\frac{\partial Y_i}{\partial X_v}$, $\frac{\partial Y_i}{\partial h_i}$ y R_i. El cálculo de cada uno de estos parámetros se detalla a continuación:

1. **Cálculo de $\frac{\partial \mathbf{Y_i}}{\partial \mathbf{h_i}}$.**

   Para obtener este jacobiano, se procede de la siguiente manera. Teniendo en cuenta la ecuación de predicción para marcas con parametrización 3D:

$$\mathbf{h_i}_{[\mathbf{3,1}]} = R^{CW} \cdot (Y_i - X_{cam}) \tag{6.100}$$

   Se puede obtener el jacobiano $\frac{\partial h_i}{\partial Y_i}$ derivando en la expresión anterior:

$$\frac{\partial \mathbf{h_i}}{\partial \mathbf{Y_i}}_{[\mathbf{3,3}]} = R^{CW} \tag{6.101}$$

   Si tenemos en cuenta que el jacobiano anterior es invertible, podemos obtener el jacobiano buscado:

$$\frac{\partial \mathbf{Y_i}}{\partial \mathbf{h_i}}_{[\mathbf{3,3}]} = \left(\frac{\partial h_i}{\partial Y_i}\right)^t = R^{WC} \tag{6.102}$$

2. **Cálculo de $\frac{\partial \mathbf{Y_i}}{\partial \mathbf{X_v}}$.**

   En este caso, en primer lugar se obtendrá el jacobiano $\frac{\partial Y_i}{\partial X_p}$. Para ello, el primer paso consiste en calcular el jacobiano $\frac{\partial h_i}{\partial X_p}$.

   Dado que el vector X_p está compuesto por el vector de posición X_{cam} y el vector de rotación q_{cam}, dicho jacobiano puede dividirse en dos partes:

$$\frac{\partial \mathbf{h_i}}{\partial \mathbf{X_p}}_{[\mathbf{3,7}]} = \left(\frac{\partial h_i}{\partial X_{cam}} \, , \, \frac{\partial h_i}{\partial q_{cam}}\right)^t \tag{6.103}$$

   Para el cálculo de $\frac{\partial h_i}{\partial X_{cam}}$ se puede obtener fácilmente a partir de la ecuación 6.100:

$$\frac{\partial \mathbf{h_i}}{\partial \mathbf{X_{cam}}}_{[\mathbf{3,3}]} = -R^{CW} \tag{6.104}$$

   Para la obtención del jacobiano $\frac{\partial h_i}{\partial q_{cam}}$ se necesita tener en cuenta la relación existente entre el vector de rotación q_{cam} y la matriz de rotación R^{WC}. Es decir:

$$\mathbf{R^{WC}}_{[\mathbf{3,3}]} = \begin{pmatrix} q_0^2+q_x^2-q_y^2-q_z^2 & 2\left(q_xq_y-q_0q_z\right) & 2\left(q_xq_z+q_0q_y\right) \\ 2\left(q_xq_y+q_0q_z\right) & q_0^2-q_x^2+q_y^2-q_z^2 & 2\left(q_yq_z-q_0q_x\right) \\ 2\left(q_xq_z-q_0q_y\right) & 2\left(q_yq_z+q_0q_x\right) & q_0^2-q_x^2-q_y^2+q_z^2 \end{pmatrix} \tag{6.105}$$

A partir de la ecuación 6.100, podemos obtener la siguiente expresión:

$$\frac{\partial \mathbf{h_i}}{\partial \mathbf{q_{cam}}}_{[\mathbf{3,4}]} = \frac{\partial R^{CW}}{\partial q_{cam}^{-1}} \cdot (Y_i - X_{cam}) \cdot \frac{\partial q_{cam}^{-1}}{\partial q_{cam}} \tag{6.106}$$

El cálculo de los jacobianos necesarios es el siguiente:

$$\frac{\partial \mathbf{R^{CW}}}{\partial \mathbf{q_{cam}}}_{[\mathbf{3,12}]} = \left\{ \frac{\partial R^{CW}}{\partial q_0}, \frac{\partial R^{CW}}{\partial q_x}, \frac{\partial R^{CW}}{\partial q_y}, \frac{\partial R^{CW}}{\partial q_z} \right\} \tag{6.107}$$

$$\frac{\partial \mathbf{q_{cam}^{-1}}}{\partial \mathbf{q_{cam}}}_{[\mathbf{4,4}]} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{6.108}$$

$$\frac{\partial \mathbf{R^{CW}}}{\partial \mathbf{q_0}}_{[\mathbf{3,3}]} = 2 \cdot \begin{pmatrix} q_0 & q_z & -q_y \\ -q_z & q_0 & q_x \\ q_y & -q_x & q_0 \end{pmatrix} \tag{6.109}$$

$$\frac{\partial \mathbf{R^{CW}}}{\partial \mathbf{q_x}}_{[\mathbf{3,3}]} = 2 \cdot \begin{pmatrix} q_x & q_y & q_z \\ q_y & -q_x & q_0 \\ q_z & -q_0 & -q_x \end{pmatrix} \tag{6.110}$$

$$\frac{\partial \mathbf{R^{CW}}}{\partial \mathbf{q_y}}_{[\mathbf{3,3}]} = 2 \cdot \begin{pmatrix} -q_y & q_x & -q_0 \\ q_x & q_y & q_z \\ q_0 & q_z & -q_y \end{pmatrix} \tag{6.111}$$

$$\frac{\partial \mathbf{R^{CW}}}{\partial \mathbf{q_z}}_{[\mathbf{3,3}]} = 2 \cdot \begin{pmatrix} -q_z & q_0 & q_x \\ -q_0 & -q_z & q_y \\ q_x & q_y & q_z \end{pmatrix} \tag{6.112}$$

Para obtener $\frac{\partial Y_i}{\partial X_p}$ es posible a partir de los jacobianos $\frac{\partial h_i}{\partial X_p}$ y $\frac{\partial Y_i}{\partial h_i}$ (calculado en el apartado anterior).

$$\frac{\partial \mathbf{Y_i}}{\partial \mathbf{X_p}}_{[\mathbf{3,7}]} = \frac{\partial Y_i}{\partial h_i} \cdot \frac{\partial h_i}{\partial X_p} \tag{6.113}$$

Una vez calculado $\frac{\partial Y_i}{\partial X_p}$ se puede obtener $\frac{\partial Y_i}{\partial X_v}$ de la siguiente forma:

$$\frac{\partial \mathbf{Y_i}}{\partial \mathbf{X_v}}_{[\mathbf{3,13}]} = \frac{\partial Y_i}{\partial X_p} \cdot \frac{\partial X_p}{\partial X_v} \tag{6.114}$$

Finalmente, la obtención de $\frac{\partial X_p}{\partial X_v}$ es sencilla, ya que será igual a 1 para todos los términos de derivadas parciales con respecto a sí mismos, y 0 para el resto de términos:

$$\frac{\partial \mathbf{X_p}}{\partial \mathbf{X_v}}_{[7,13]} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 \end{pmatrix} \tag{6.115}$$

3. **Cálculo de $\mathbf{R_i}$.**

   La matriz de ruido de medida expresa la incertidumbre asociada a la hora de realizar una medida de una posición 3D a través de unas proyecciones 2D. El vector de medida, está formado por las coordenadas de las diferentes marcas respecto al sistema de referencia de la cámara izquierda h_i. Debido a que la obtención del vector de observación se realiza de forma indirecta, no se puede predecir a priori dicha matriz de ruido.

   La observación parte de la medida de las coordenadas pixélicas (u_L, v_L), (u_R, v_R) correspondientes a cada una de las marcas detectadas por el sistema. A priori, se puede suponer que la incertidumbre, a la hora de realizar dicha medida, se basa en la indeterminación de saber si la proyección de la marca se encuentra en un píxel o en el adyacente.

   A partir de aquí, se puede suponer que la incertidumbre en la medida es de ± 1 píxel. Además de esto, se puede suponer que dicha indeterminación es independiente tanto en la coordenada u como en la coordenada v (ver figura 6.24).

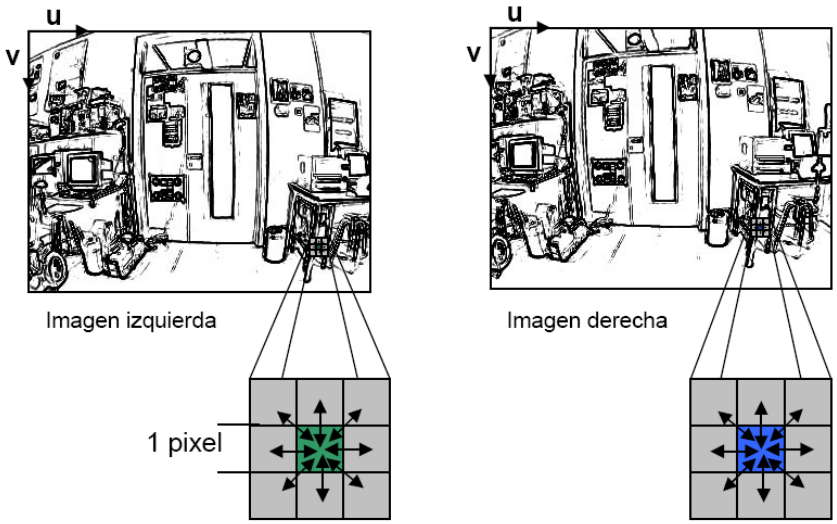


Figura 6.24: Incertidumbre en la medida pixélica

En base a estas suposiciones, se puede construir un vector formado por los pates de coordenadas de imagen de cada marca:

$$\mathbf{T_i}_{[4,1]} = (u_L \; v_L \; u_R \; v_R)^t \tag{6.116}$$

Si tratamos a las 4 variables del vector T_i como variables aleatorias de tipo Gaussiano con media cero y desviación típica de valor 1 píxel ($\sigma = 1 \; pxel$). Por lo tanto, es posible definier una matriz de ruido independiente a partir de las 4 variables aleatorias:

$$\mathbf{R_{T_i}}_{[4,4]} = \begin{pmatrix} \sigma_{u_L}^2 & 0 & 0 & 0 \\ 0 & \sigma_{v_L}^2 & 0 & 0 \\ 0 & 0 & \sigma_{u_R}^2 & 0 \\ 0 & 0 & 0 & \sigma_{v_R}^2 \end{pmatrix} \tag{6.117}$$

Para poder calcular la matriz R_i correspondiente al vector de observación h_i se deberá realizar la siguiente transformación:

$$\mathbf{R_i}_{[3,3]} = \frac{\partial h_i}{\partial T_i} \cdot R_{T_i} \cdot \left(\frac{\partial h_i}{\partial T_i}\right)^t \tag{6.118}$$

El problema ahora consiste en obtener el valor del jacobiano $\frac{\partial h_i}{\partial T_i}$. Para ello partimos de la ecuación para obtener las coordenadas 3D de cada marca a partir de las proyecciones pixélicas en cada una de las cámaras (ver sección 4.2.3):

$$A_i \cdot h_i = b_i \tag{6.119}$$

Para poder obtener las matrices A y b es necesario conocer las coordenadas de proyección (u_L, v_L) y (u_R, v_R). El siguiente paso es calcular los jacobianos de los vectores a ambos lados de la ecuación:

$$\frac{\partial A_i \cdot h_i}{\partial T_i} = \frac{\partial b_i}{\partial T_i} \tag{6.120}$$

Desarrollando los jacobianos anteriores se puede obtener un sistema de 12 ecuaciones con 12 incógnitas. De estas ecuaciones se obtienen los 12 elementos del jacobiano buscado $\frac{\partial h_i}{\partial T_i}$. Reagrupando las ecuaciones, se puede expresar el sistema de la siguiente forma:

$$A_i \cdot \frac{\partial h_i}{\partial T_i} = C_i \tag{6.121}$$

en donde la matriz C_i tiene la siguiente expresión:

$$\mathbf{C_i}_{[4,4]} = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & b & 0 \\ 0 & 0 & 0 & b \end{pmatrix} \tag{6.122}$$

$$\begin{cases} \mathbf{a} = -m_{L_{31}} h_{ix} - m_{L_{32}} h_{iy} - m_{L_{33}} h_{iz} - m_{L_{34}} \\ \mathbf{b} = -m_{R_{31}} h_{ix} - m_{R_{32}} h_{iy} - m_{R_{33}} h_{iz} - m_{R_{34}} \end{cases} \tag{6.123}$$

Por lo tanto, se obtiene el jacobiano $\frac{\partial h_i}{\partial T_i}$ como sigue:

$$\frac{\partial \mathbf{h_i}}{\partial \mathbf{T_i}}_{[3,4]} = \left(A_i^t \cdot A_i\right)^{-1} \cdot A_i^t \cdot C_i \tag{6.124}$$

6.9.2. Marcas con Parametrización Inversa

Los diferencias existentes entre la parametrización inversa y la parametrización 3D implica que el único jacobiano diferente entre ambos sea $\frac{\partial h_i}{\partial Y_i}$. El proceso para calcular el resto de jacobianos, es el mismo que en la sección anterior cambiando únicamente las dimensiones de los mismos. A continuación, se exponen los jacobianos necesarios para el cálculo de $\frac{\partial h_i}{\partial Y_i}$.

Sea la ecuación de predicción para una marca con parametrización inversa:

$$\mathbf{h_i}_{\,[\mathbf{3},\mathbf{1}]} = R^{CW} \cdot \left((X_{ori} - X_{cam}) + \frac{1}{\rho} \cdot m\left(\theta, \phi\right) \right) \tag{6.125}$$

en donde como se explicó en la sección 6.3.2, el vector unitario $m(\theta, \phi)$ presenta los siguientes valores:

$$\mathbf{m}(\theta, \phi)_{\,[\mathbf{3},\mathbf{1}]} = (\sin\phi_i \cos\theta_i, -\cos\theta_i, \sin\phi_i \sin\theta_i)^t \tag{6.126}$$

A partir de las ecuaciones de predicción y la del vector unitario $m(\theta, \phi)$ se pueden obtener las expresiones de los jacobianos necesarios:

$$\frac{\partial \mathbf{h_i}}{\partial \mathbf{Y_i}}_{\,[\mathbf{3},\mathbf{6}]} = \left(\frac{\partial h_i}{\partial X_{ori}}, \frac{\partial h_i}{\partial \theta_i}, \frac{\partial h_i}{\partial \phi_i}, \frac{\partial h_i}{\partial 1/\rho_i} \right)^t \tag{6.127}$$

$$\frac{\partial \mathbf{h_i}}{\partial \mathbf{X_{ori}}}_{\,[\mathbf{3},\mathbf{3}]} = R^{CW} \tag{6.128}$$

$$\frac{\partial \mathbf{h_i}}{\partial \mathbf{1}/\rho_\mathbf{i}}_{\,[\mathbf{3},\mathbf{1}]} = R^{CW} \cdot m\left(\theta, \phi\right) \tag{6.129}$$

$$\frac{\partial \mathbf{h_i}}{\partial \theta_\mathbf{i}}_{\,[\mathbf{3},\mathbf{1}]} = \frac{R^{CW}}{\rho_i} \cdot \begin{pmatrix} -\sin\phi_i \sin\theta_i \\ 0 \\ \sin\phi_i \cos\theta_i \end{pmatrix} \tag{6.130}$$

$$\frac{\partial \mathbf{h_i}}{\partial \phi_\mathbf{i}}_{\,[\mathbf{3},\mathbf{1}]} = \frac{R^{CW}}{\rho_i} \cdot \begin{pmatrix} \cos\phi_i \sin\theta_i \\ \sin\phi_i \\ \cos\phi_i \sin\theta_i \end{pmatrix} \tag{6.131}$$

6.10. Adaptación del Vector de Estado Global y su Covarianza

Una vez obtenidos todos los elementos necesarios para formar la nueva matriz de covarianza P, es posible construir dicha matriz de la forma que se explica a continuación. Sea la marca nueva a añadir Y_n, esta marca será inicializada con lo siguientes valores de covarianzas:

$$P_{Y_n X} = \frac{\partial Y_n}{\partial X_v} \cdot P_{XX} \tag{6.132}$$

$$P_{Y_n Y_i} = \frac{\partial Y_n}{\partial X_v} \cdot P_{XY_i} \tag{6.133}$$

$$P_{Y_nY_n} = \frac{\partial Y_n}{\partial X_v} \cdot P_{XX} \cdot \left(\frac{\partial Y_n}{\partial X_v}\right)^t + \frac{\partial Y_n}{\partial h_n} \cdot R_n \cdot \left(\frac{\partial Y_n}{\partial h_n}\right)^t \tag{6.134}$$

Y su adaptación a la matriz de covarianza del mapa quedaría como sigue:

$$P = \begin{pmatrix} P_{XX} & P_{XY_1} & \cdots & P_{XX} \cdot \left(\frac{\partial Y_n}{\partial X_v}\right)^t \\ P_{Y_1X} & P_{Y_1Y_1} & \cdots & P_{Y_1X} \cdot \left(\frac{\partial Y_n}{\partial X_v}\right)^t \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial Y_n}{\partial X_v} \cdot P_{XX} & \frac{\partial Y_n}{\partial X_v} \cdot P_{XY_1} & \cdots & \frac{\partial Y_n}{\partial X_v} \cdot P_{XX} \cdot \left(\frac{\partial Y_i}{\partial X_v}\right)^t + \frac{\partial Y_n}{\partial h_n} \cdot R_n \cdot \left(\frac{\partial Y_n}{\partial h_n}\right)^t \end{pmatrix} \tag{6.135}$$

Es decir, se trata de añadir una nueva columna por la derecha y una nueva fila por la parte inferior en base a los elementos obtenidos en la sección 6.9. Cabe destacar, que en el caso de que las marcas nuevas a añadir presenten una parametrización 3D o una parametrización inversa, lo único que cambia es el tamaño de las covarianzas anteriores.

La adaptación del vector de estado es sencilla, ya que simplemente hay que añadir el vector de estado de la nueva marca como nueva fila en el vector de estado global

$$X = \begin{pmatrix} X_v \\ Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \tag{6.136}$$

6.11. Eliminación de Marcas

A la hora de eliminar una marca en el proceso del filtro, bastará con eliminar su fila y columna correspondiente en la matriz de covarianza total P. Por ejemplo, eliminando del proceso del filto la marca número 1:

$$P = \begin{pmatrix} P_{XX} & P_{XY_1} & P_{XY_2} & \cdots & P_{XY_n} \\ P_{Y_1X} & P_{Y_1Y_1} & P_{Y_1Y_2} & \cdots & P_{Y_1Y_n} \\ P_{Y_2X} & P_{Y_2Y_1} & P_{Y_2Y_2} & \cdots & P_{Y_2Y_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{Y_nX} & P_{Y_nY_1} & P_{Y_nY_2} & \cdots & P_{Y_nY_n} \end{pmatrix} \Rightarrow P = \begin{pmatrix} P_{XX} & P_{XY_2} & \cdots & P_{XY_n} \\ P_{Y_2X} & P_{Y_2Y_2} & \cdots & P_{Y_2Y_n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{Y_nX} & P_{Y_nY_2} & \cdots & P_{Y_nY_n} \end{pmatrix} \tag{6.137}$$

Respecto al vector de estado, simplemente será necesario eliminar la marca de su posición correspondiente.

$$X = \begin{pmatrix} X_v \\ Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \Rightarrow X = \begin{pmatrix} X_v \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \tag{6.138}$$

6.12. Conmutación entre Parametrización Iversa y 3D

Una vez que se ha realizado la predicción de las marcas, para aquellas marcas visibles en ese instante de tiempo, hay que determinar si es necesario cambiar o no la parametrización original de la marca. Para ello, si una marca con parametrización 3D presenta una profundidad mayor al umbral establecido para conmutar (ver sección 6.3.3.3) será necesario parametrizar la marca con una parametrización inversa, y viceversa. Además, se impone una restricción antes de realizar la conmutación y la adaptación de la marca, y es que la marca debe haber permanecido al menos 5 frames consecutivos en su nuevo estado de parametrización. Una vez que se supere este número de frames, se realiza la adaptación del vector de estado de la marca y de las covarianzas implicadas. El motivo de esperar un número de frames determinado antes de realizar la conmutación completa es para evitar conversiones muy seguidas entre ambas parametrizaciones debido a posibles errores en la medida de predicción.

El vector de estado de la marca y el jacobiano $\frac{\partial h_i}{\partial Y_{INV}}$ se calculan como se ha visto en la sección 6.9. Sin embargo, es necesario realizar una adaptación de las covarianzas implicadas en el proceso. Según los distintos casos posibles de conmutación, la adaptación de las covarianzas implicadas es la siguiente:

- Una marca con parametrización 3D pasa a tener una parametrización inversa.

$$\mathbf{P_{YY_{INV}\ [6,6]}} = \left(\frac{\partial h_i}{\partial Y_{INV}}\right)^t \cdot P_{YY_{3D}} \cdot \left(\frac{\partial h_i}{\partial Y_{INV}}\right) \tag{6.139}$$

$$\mathbf{P_{XY_{INV}\ [13,6]}} = P_{XY_{3D}} \cdot \left(\frac{\partial h_i}{\partial Y_{INV}}\right) \tag{6.140}$$

- Una marca con parametrización inversa pasa a tener una parametrización 3D.

$$\mathbf{P_{YY_{3D}\ [3,3]}} = \left(\frac{\partial h_i}{\partial Y_{INV}}\right) \cdot P_{YY_{INV}} \cdot \left(\frac{\partial h_i}{\partial Y_{INV}}\right)^t \tag{6.141}$$

$$\mathbf{P_{XY_{3D}\ [13,3]}} = P_{XY_{INV}} \cdot \left(\frac{\partial h_i}{\partial Y_{INV}}\right)^t \tag{6.142}$$

Hay que tener en cuenta que en el momento que se cambia una marca de parametrización en el proceso, es necesario adaptar a su vez todas las covarianzas cruzadas de las marcas en el mapa con respecto de la marca que ha cambiado. Se distinguen los siguientes casos:

- Sean la marca j una marca con parametrización 3D y que pasa a tener una parametrización inversa, y la marca i que presenta una parametrización 3D.

$$\mathbf{P_{Y_{i\ 3D}Y_{j\ INV}\ [3,6]}} = P_{Y_{i\ 3D}Y_{j\ 3D}} \cdot \left(\frac{\partial h_j}{\partial Y_{INV}}\right) \tag{6.143}$$

- Sean la marca j una marca con parametrización inversa y que pasa a tener una parametrización 3D, y la marca i que presenta una parametrización 3D.

$$\mathbf{P_{Y_{i\ 3D}Y_{j\ 3D}\ [3,3]}} = P_{Y_{i\ 3D}Y_{j\ INV}} \cdot \left(\frac{\partial h_j}{\partial Y_{INV}}\right)^t \tag{6.144}$$

- Sean la marca j una marca con parametrización 3D y que pasa a tener una parametrización inversa, y la marca i que presenta una parametrización inversa.

$$\mathbf{P_{Y_i\ INV Y_j\ INV}}_{[\mathbf{6},\mathbf{6}]} = P_{Y_i\ INV Y_j\ 3D} \cdot \left(\frac{\partial h_j}{\partial Y_{INV}} \right) \tag{6.145}$$

- Sean la marca j una marca con parametrización inversa y que pasa a tener una parametrización 3D, y la marca i que presenta una parametrización inversa.

$$\mathbf{P_{Y_i\ INV Y_j\ 3D}}_{[\mathbf{6},\mathbf{3}]} = P_{Y_i\ INV Y_j\ INV} \cdot \left(\frac{\partial h_j}{\partial Y_{INV}} \right)^t \tag{6.146}$$

Capítulo 7

Resultados

En este capítulo se analizan los resultados experimentales del sistema bajo varias secuencias de prueba realizadas todas ellas en escenarios de interiores, en la Escuela Politécnica Superior de la Universidad de Alcalá.

Inicialmente se comentarán las características principales de cada uno de los vídeos de prueba utilizados. Posteriormente se realizará un estudio y comparativa de los distintos métodos utilizados, para finalizar con un apartado sobre los posibles errores del sistema.

Dentro del análisis de la comparativa de los distintos métodos utilizados, los estudios que se han realizado son los siguientes:

- **Comparación entre Detectores de Marcas**: En este apartado se realiza una comparativa entre los 4 detectores estudiados: Shi-Tomasi, Harris, Afín Invariante y Diferencia de Gaussianas. Para hacer la comparativa, se muestran resultados de los tiempos de ejecución medios de cada detector, del número de intentos de medida de marcas, del número de intentos de medida de marcas correctos, del número de marcas del mapa final, así como de la reconstrucción de la trayectoria seguida por la cámara.

- **Comparación entre Parametrización 3D y Parametrización Inversa**: En este apartado se propone comparar la reconstrucción de un mapa 3D utilizando únicamente la parametrización 3D, y la reconstrucción del mapa considerando ambas parametrizaciones de las marcas. Además se realiza también una comparación entre distintos umbrales de profundidad para conmutar entre marcas 3D o inversas.

- Comparación entre distintos Métodos de Adaptación de Parches: En este apartado se realiza una comparativa entre los diferentes métodos de adaptación de parches estudiados: Parche Adaptado y Warping mediante Homografía. También se realiza una comparación sin realizar adaptación de parches.

Se han utilizado tres secuencias de prueba en interiores. Las principales características de cada una de las secuencias utilizadas son:

- **Secuencia 1: Pasillo.** Esta secuencia muestra un pasillo de aproximadamente 8 m de longitud. A través de esta secuencia podremos comprobar como se comporta el sistema, cuando se realiza una trayectoria recta. El número de frames de esta secuencia es de 800. En la figura 7.1(a) se muestra una imagen de la secuencia 1.

- **Secuencia 2: L.** Esta secuencia muestra una trayectoría recta de aproximadamente 6 m, para posteriormente realizar un giro a la derecha de aproximadamente 3 m. El número de frames de esta secuencia es de 840. En la figura 7.1(b) se muestra una imagen de la secuencia 2.

(a) Secuencia 1

(b) Secuencia 2

Figura 7.1: Imágenes de las Secuencias de Test 1 y 2

- **Secuencia 3: Loop.** Esta secuencia muestra un bucle de aproximadamente 4.8 m a lo largo del eje X, y de aproximadamente 6 m a lo largo del eje Z. El número de frames de esta secuencia es de 1328. En la figura 7.1(b) se muestra una imagen de la secuencia 3.
- **Secuencia 4: Pasillo.** Esta secuencia muestra un pasillo de aproximadamente 10 m de longitud. El número de frames de esta secuencia es de 600. En la figura 7.2(b) se muestra una imagen de la secuencia 4.

(a) Secuencia 3

(b) Secuencia 4

Figura 7.2: Imágenes de las Secuencias de Test 3 y 4

Posteriormente, se representa el mapa 3D de cada secuencia, generado con cada uno de los distintos detectores utilizados. Además, se muestra una tabla en la que se presenta una estimación métrica de la trayectoria realizada.

En las secuencias anteriores, cabe destacar que se realizaron considerando una velocidad de una persona humana caminando ($3\ Km/h\ -\ 4\ Km/h$) con una cámara estéreo movida por lo mano. También, es necesario mencionar que aunque el proceso de Visual SLAM es un proceso aleatorio, la repetibilidad de los resultados puede no ser exactamente la misma, existiendo una

pequeña desviación, sin embargo, las conclusiones que se exponen en la sección 7.1.4 si que se pueden generalizar.

Los siguientes resultados se han obtenido utilizando un procesador Intel Core 2 Duo funcionando a 2.4 GHz.

7.1. Comparación entre Detectores de Marcas

A continuación se muestran las tablas de resultados del análisis de los distintos detectores de marcas naturales estudiados. Para ello se analizan una serie de parámetros que aparecen en las tablas, estos son:

- **Detector**: Este campo indica el nombre del detector de marcas utilizado.
- **Tiempo**: Este campo indica el tiempo medio en ms que supone una iteración del detector utilizado
- **# Marcas Mapa**: Este campo indica el número total de marcas necesarias para la obtención del mapa 3D del entorno.
- **# Total**: Este campo indica el número total de intentos de medida de marcas a lo largo de toda la secuencia.
- **# Correctos**: Este campo indica el número de intentos de medida de marcas que han resultado satisfactorios a lo largo de la secuencia.
- **Ratio**: Este campo indica el ratio en porcentaje entre el número de intentos de medida correctos y el número de intentos de medida total.

En lo que respecta a este primer estudio, **no se ha considerado ninguna adaptación de parches**, **ni la parametrización inversa de las marcas**.

7.1.1. Secuencia 1

Detector	Tiempo (ms)	# Marcas Mapa	# Total	# Correctos	Ratio %
Shi-Tomasi	9.27	92	6230	5738	92.10
Harris	5.68	82	6539	5934	90.74
Afín Invariante	23.7	69	6119	5593	91.40
DOG	85.25	75	6276	5731	91.32

Tabla 7.1: Comparativa de Intentos de Medida de Marcas: Secuencia 1

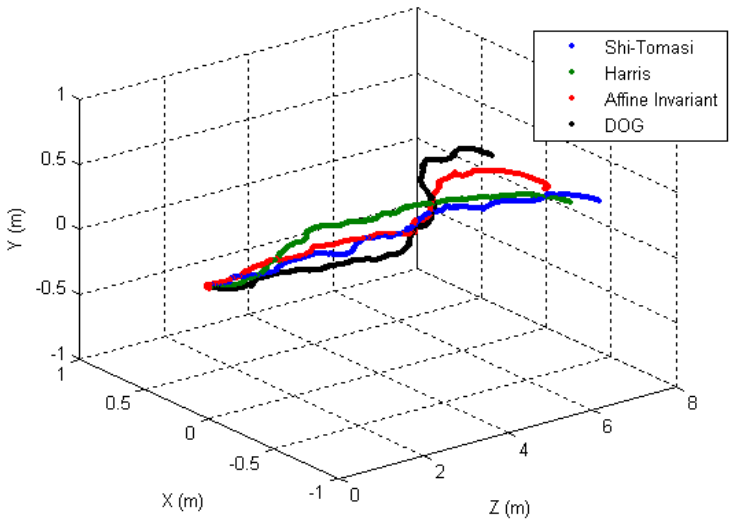


Figura 7.3: Mapa 3D de la Secuencia 1

Detector	ΔX (m)	ΔY (m)	ΔZ (m)
Shi-Tomasi	0.3895	0.1156	8.2563
Harris	0.5166	0.0931	7.8741
Afín Invariante	0.4232	0.1345	7.9921
DOG	0.8950	0.0968	8.1614

Tabla 7.2: Comparativa de Estimaciones de Trayectoria Secuencia 1

7.1.2. Secuencia 2

Detector	Tiempo (ms)	# Marcas Mapa	# Total	# Correctos	Ratio %
Shi-Tomasi	9.27	115	11150	9117	81.76
Harris	5.68	114	10355	9135	88.21
Afín Invariante	23.7	105	9686	8881	91.68
DOG	85.25	100	9741	9022	92.61

Tabla 7.3: Comparativa de Intentos de Medida de Marcas: Secuencia 2

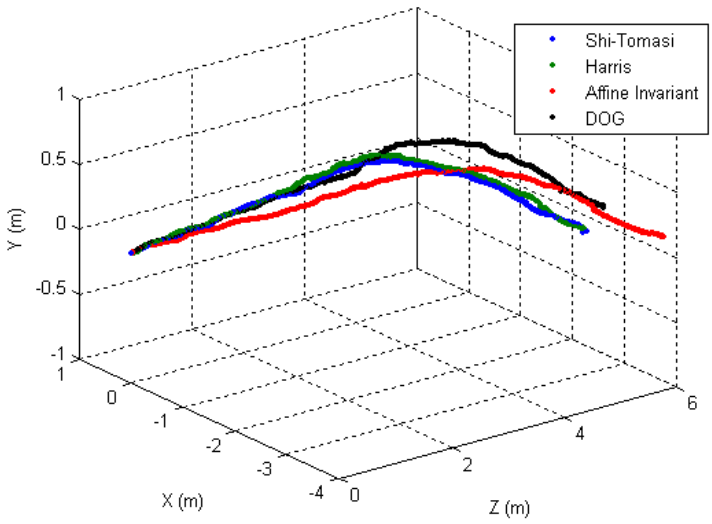


Figura 7.4: Mapa 3D de la Secuencia 2

Detector	ΔX (m)	ΔY (m)	ΔZ (m)
Shi-Tomasi	3.6022	0.1768	5.1980
Harris	3.5831	0.1552	5.4010
Afín Invariante	3.5081	0.0703	6.3071
DOG	3.2037	0.1353	6.2499

Tabla 7.4: Comparativa de Estimaciones de Trayectoria Secuencia 2

7.1.3. Secuencia 3

Detector	Tiempo (ms)	# Marcas Mapa	# Total	# Correctos	Ratio %
Shi-Tomasi	9.27	207	16430	15519	94.45
Harris	5.68	192	16301	15475	94.93
Afín Invariante	23.7	182	15415	14742	95.64
DOG	85.25	198	15237	14603	95.83

Tabla 7.5: Comparativa de Intentos de Medida de Marcas: Secuencia 3

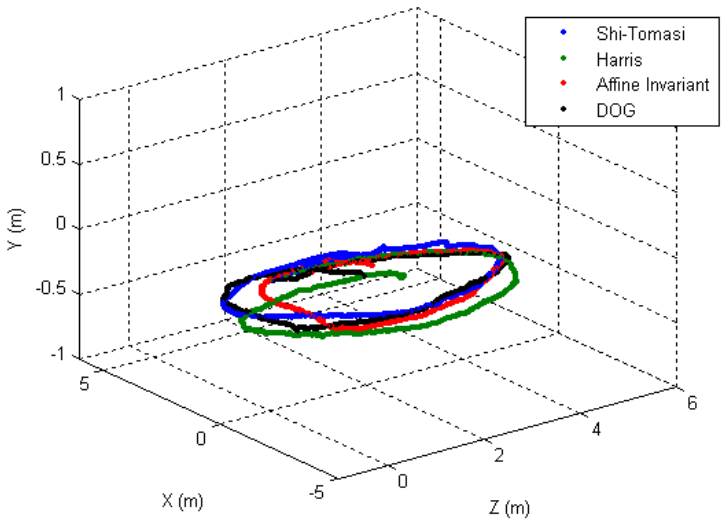


Figura 7.5: Mapa 3D de la Secuencia 3

Detector	ΔX (m)	ΔY (m)	ΔZ (m)
Shi-Tomasi	4.7190	0.1048	4.9979
Harris	5.3335	0.1259	5.0058
Afín Invariante	5.0851	0.1036	4.1805
DOG	5.0972	0.1184	4.7544

Tabla 7.6: Comparativa de Estimaciones de Trayectoria Secuencia 3

7.1.4. Conclusiones

Observando los resultados, se puede llegar a las siguientes conclusiones sobre los distintos tipos de detectores de marcas estudiados:

- Desde el punto de vista de **tiempo de cómputo medio por ejecución**: Como se comentó en el capítulo 5, el detector DOG presenta un tiempo de cómputo prohibitivo para una aplicación cuyo objetivo sea funcionar en tiempo real. El detector afín invariante presenta un tiempo de cómputo algo elevado para aplicaciones de tiempo real, sin embargo, hay que tener en cuenta que la ejecución del detector de marcas no se realiza cada frame, si no solamente cuando es necesario añadir una nueva marca al proceso. Del resto de detectores, el que menor tiempo de cómputo presenta es el detector de esquinas de Harris, siendo su tiempo de cómputo muy pequeño, prácticamente la mitad que el detector de esquinas de Shi-Tomasi.

- Desde el punto de vista del **número de marcas por mapa**: De manera general, con el detector afín invariante se obtienen mapas 3D con un menor número de marcas, ya que se trata de un detector de marcas más selectivo que el resto, sólo detectando marcas aquellos puntos en la imagen donde la curvatura de las isolíneas es elevada. Aunque no obstante, la diferencia con respecto al resto de detectores no es elevada. Esta diferencia en el número de marcas por mapa, podría tener su importancia en una implementación de más alto nivel, en la que se tuviera un mapa global y varios submapas.

- Desde el punto de vista del **número de intentos totales de medida**: Este parámetro es indicativo a su vez del número de marcas totales que se han añadido al sistema. Conviene aclarar en este punto que **el número de marcas por mapa** es el **número de marcas finales de la que consta el mapa**, pero para llegar a este mapa final se habrán añadido otra serie de marcas que posteriormente han sido eliminadas del proceso por no haber sido marcas idóneas para realizar un seguimiento. En media, el detector que más marcas introduce al sistema y que por tanto más intentos de medida realiza, es el detector de Shi-Tomasi.

- Desde el punto de vista del **número de intentos de medida correctos** y del **ratio** entre correctas y medidas totales, este parámetro es bastante alto en todos los detectores. Esto es debido a que para que el algoritmo coverga y se obtengan mapas precisos, es necesario disponer de al menos 10 marcas visibles y medidas correctamente en cada iteración, si alguna de las anteriores condiciones no se cumple, el sistema intenta añadir una nueva marca, y además aquellas marcas cuyo ratio entre número de medidas correctas y número de intentos de medida es menor que el 70 % son eliminadas del mapa. El detector que presenta unos ratios de acierto más elevados es el DOG.

- Desde el punto de vista de la estimación del **mapa 3D** , se obtienen resultados similares con todos los detectores, cometiendo un pequeño error con respecto al caso real.

Los resultados experimentales muestran que con cualquiera de los detectores estudiados, se pueden obtener buenos mapas 3D, siempre que se encuentren ajustados correctamente los parámetros de cada uno de los detectores. Para concluir el estudio, se propone elegir el **detector de Harris** como el mejor para la aplicación teniendo en cuenta las restricciones de la misma, ya que es aquel que presenta un menor tiempo de cómputo, presenta unos ratios de detección correcta elevados, y el número final de marcas por mapa es aceptable.

7.2. Comparación entre Parametrización 3D y Parametrización Inversa

En esta sección se muestra una comparativa de resultados considerando los dos tipos de parametrizaciones de marcas implementados. Debido a que de momento solamente se ha trabajado con secuencias de interiores, se ha escogido la secuencia 4 (pasillo) como secuencia de prueba para realizar este análisis, ya que en interiores el número de marcas que podrían ser susceptibles de tener una parametrización inversa, es menor que en entornos de exteriores.

Para la siguiente comparación no se ha utilizado ninguna adaptación de parches. Se comparan los resultados obtenidos sin considerar la parametrización inversa, y considerando ambas parametrizaciones pero con dos umbrales de profundidad distintos para conmutar entre marcas inversas y 3D. Dichos umbrales son $z_1 = 5{,}71\ m$ y $z_2 = 10\ m$.

Como se explicó en la sección 6.3, al utilizar la parametrización inversa de las marcas, el tamaño del vector de estado de una marca tiene dimensión 6×1, mientras que utilizando una parametrización 3D, el tamaño del vector de estado de una marca es de 3×1. Esta diferencia de tamaños puede ser importante en secuencias donde el número de marcas inversas sea muy elevado en comparación con el número de marcas 3D. En la figura 7.6 se observa una comparativa del tamaño del vector de estado en función del número de frames de la secuencia 4.

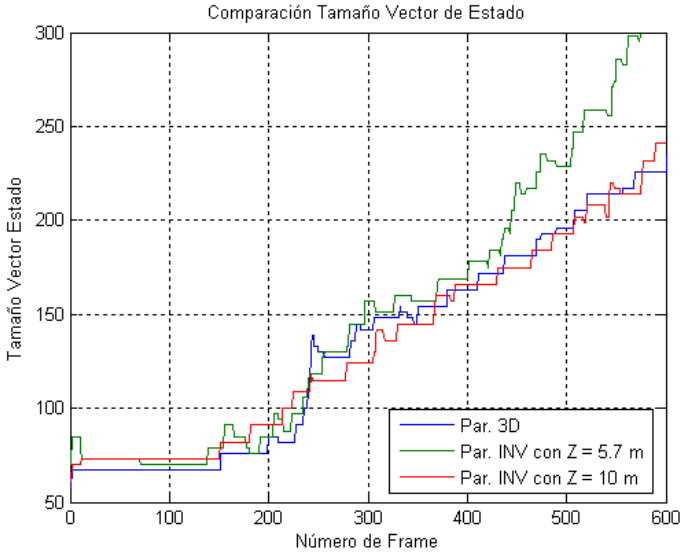


Figura 7.6: Comparativa Tamaño del Vector de Estado

Como se aprecia en la figura 7.6 la diferencia en el tamaño del vector de estado es considerable a medida que aumenta el número de frames, siendo el tamaño del vector de estado más elevado para el caso en el que utilicemos una parametrización inversa con un umbral $z_1 = 5{,}71\ m$. Sin embargo, comparando el tamaño que se obtiene sin utilizar la parametrización inversa y utilizándola con un umbral $z_2 = 10\ m$, esta diferencia de tamaño es menos significativa, ya que únicamente se parametrizan como marcas inversas aquellas que están verdaderamente lejanas a la cámara en donde la medida de profundidad tiene un gran error.

Esta diferencia en tamaño del vector de estado, se traduce a su vez en un coste computacional más alto a mayor tamaño del vector de estado. En la figura 7.7 se puede observar una comparativa

con respecto al tiempo de ejecución por frame del algoritmo.

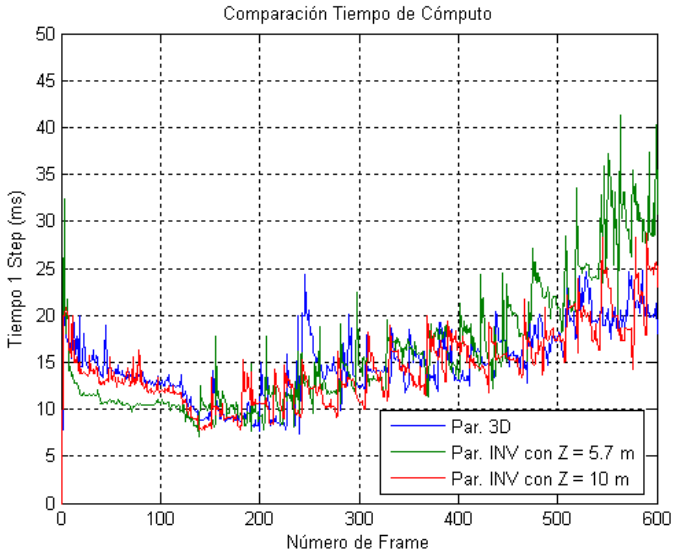


Figura 7.7: Comparativa Tiempo de Cómputo por ejecución del algoritmo

Para el caso de utilizar una parametrización inversa con un umbral $z_1 = 5{,}71\ m$ el tiempo de ejecución en los últimos frames de la secuencia es elevado superando los 30 ms. Lo que queda por comprobar, es si utilizando ese umbral de profundidad óptimo, la calidad del mapa 3D mejora en comparación con el resto de casos. En la figura 7.8 se puede observar la trayectoria recta obtenida con cada uno de los casos estudiados:

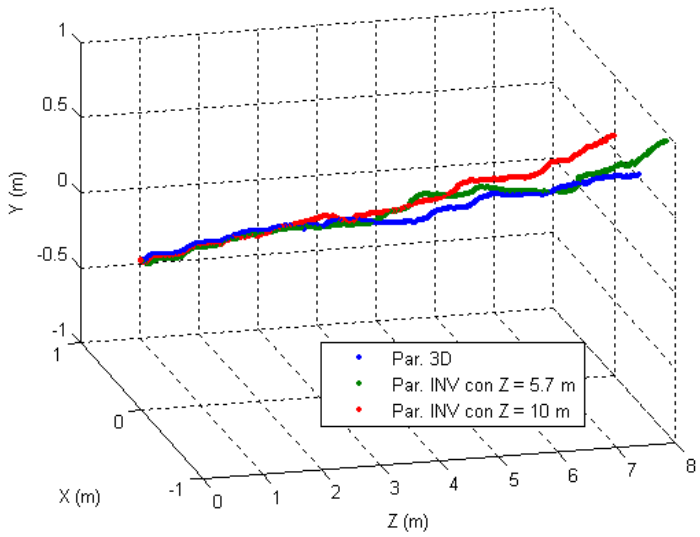


Figura 7.8: Comparativa de Estimaciones de Trayectoria

Caso	**ΔX (m)**	**ΔY (m)**	**ΔZ (m)**
Sin Par. Inversa	0.8800	0.0836	9.0624
Con Par. Inversa, $z_1 = 5{,}7\ m$	0.6962	0.1007	9.8747
Con Par. Inversa, $z_2 = 10\ m$	0.7583	0.1110	9.6210

Tabla 7.7: Comparativa de Estimaciones de Trayectoria Secuencia 4

7.2.1. Conclusiones

Como se puede apreciar en la figura 7.8 y en la tabla 7.7, la calidad del mapa 3D final no se ve gravemente alterada por el hecho de utilizar solamente la parametrización 3D o utilizar ambas parametrizaciones, con diferentes umbrales de profundidad. Aunque no obstante, por el hecho de utilizar una parametrización inversa al introducir en el vector de estado valores de ángulos en vez de valores 3D (con gran error debido a las elevadas distancias), hace que el proceso sea más lineal, por lo tanto mejor para el proceso de filtrado con el EKF.

Para realizar un estudio más detallado, sería necesario el estudio de secuencias en exteriores, y el ver como afecta utilizar una parametrización inversa en escenarios donde el número de marcas susceptibles de tener una parametrización inversa sea muy elevado. En el caso de utilizar visión estéreo, el empleo de una parametrización inversa de las marcas no es tan importante desde el punto de vista de reconstrucción del mapa, como en el caso de visión monocular, en el cuál no se puede estimar la profundidad directamente.

Para concluir, la utilización de un umbral de profundidad determinado depende del escenario de trabajo, así como de un compromiso necesario entre linearidad y tiempo de cómputo. Si se trabaja en un escenario de interiores en el cuál existen marcas muy lejanas, conviene utilizar un umbral de profundidad por ejemplo de unos 10 m, para que el tiempo de cómputo no sea excesivo y se cumplan las restricciones de tiempo real. De manera general, para un sistema estéreo el número de marcas con parametrización 3D debe ser superior al número de marcas que presentan parametrización inversa.

7.3. Comparación entre distintos Métodos de Adaptación de Parches

En esta sección se muestran los resultados de la comparativa entre los dos métodos de adaptación de parches implementados, y sin realizar ningún tipo de adaptación de parches. Las secuencias utilizadas para realizar esta comparativa, son la secuencia 1 (pasillo), para ver el comportamiento de estos algoritmos ante traslaciones y la secuencia 2 (L) para ver el comportamiento ante rotaciones. En las tablas 7.8, 7.9 se muestran los resultados del estudio comparativo:

Método	**# Marcas Mapa**	**# Total**	**# Correctos**	**Ratio %**
Sin Adaptación	85	6283	5612	89.32
Parche Adaptado	72	6347	5747	90.55
Homografía	68	6398	5781	90.35

Tabla 7.8: Comparativa de Métodos de Adaptación de Parches: Secuencia 1

Método	**# Marcas Mapa**	**# Total**	**# Correctos**	**Ratio %**
Sin Adaptación	116	11627	8922	76.73
Parche Adaptado	188	16793	8662	51.58
Homografía	105	10279	9119	88.71

Tabla 7.9: Comparativa de Métodos de Adaptación de Parches: Secuencia 2

En la secuencia 1, al tratarse de un pasillo, el resultado de los distintos métodos de adaptación es similar. Cabe destacar que el método de adaptación mediante homografía es aquel que obtiene un menor número de marcas válidas en el mapa final, y que es aquel en el que mayor número de intentos de medida de marcas se realizan, aunque la diferencia no es destacable en comparación con el resto de métodos.

Sin embargo en la secuencia 2, al realizarse un giro pronunciado, se empiezan a notar las diferencias entre los distintos métodos implementados. El método del parche adaptado falla considerablemente al realizar el giro, mientras que en pasillos o trayectorias rectas se comporta adecuadamente como en la secuencia 1. Para la secuencia 2, se obtienen mejores resultados con el método de la homografía, ya que es aquel que ofrece un mejor ratio de marcas medidas correctamente, además de ser aquel que obtiene un número menor de intentos de medida realizados.

En lo que respecta a la construcción del mapa 3D, ambos métodos ofrecen resultados similares, exceptuando el método del parche adaptado para la secuencia 2, en la cuál la estimación de la trayectoria seguida por la cámara se desvía un poco al realizar el giro con respecto de la trayectoria real.

7.4. Errores

Por último, cabe destacar algunos aspectos relativos a algunos de los distintos tipos de errores presentes en el proceso. A parte del error existente debido al modelado de la cámara, podríamos clasificar el resto de errores de la siguiente forma:

7.4.1. Acumulativos

Este tipo de errores se debe a la deriva producida en la localización progresiva de las marcas. A la hora de calcular la posición de una marca, ésta debe ser obtenida con un determinado error a partir de la posición y orientación actuales de la cámara. A su vez, la posición y la orientación de la cámara llevan asociadas una incertidumbre y un determinado error, ya que han sido calculados a partir de las posiciones de las diferentes marcas obtenidas en el instante anterior, acumulándose por tanto los errores ocurridos en ambos pasos del filtro.

Las marcas llevan asociada una incertidumbre en su posición, por lo que a la hora de revisitar estas marcas se reduciría la incertidumbre asociada y por tanto el error. Sin embargo, si el camino recorrido por la cámara es lo suficientemente grande, el error acumulado podría provocar que las mismas marcas no fueran reconocidas como tales, perdiendo así la posibilidad de reducir su propia incertidumbre.

7.4.2. Pérdida Total

Para poder localizar la proyección de las marcas, se realizan búsquedas de correlación dentro de un área de búsqueda de máxima probabilidad. Esto implica que para poder reconocer una marca, su proyección debe estar dentro de este área de búsqueda, lo que implica que es necesario que la posición relativa de dicha proyección no difiera demasiado entre un frame y su sucesivo. Este límite viene impuesto por la región de búsqueda anteriormente descrita.

Partiendo de lo anteriormente expuesto, si superamos una determinada aceleración lineal o angular en el movimiento de la cámara, puede ocurrir que las predicciones de las proyecciones de todas las marcas visibles en ese momento caigan fuera de sus respectivas áreas de búsqueda. En ese momento se habrá perdido toda referencia para localizar a la cámara en su entorno, pasando a un estado de **pérdida total** de localización, siendo necesario una reinicialización de la posición inicial.

Para evitar posibles estados de pérdida total, es necesario cumplir con la restricción de que el movimiento de la cámara debe ser suave, sin aceleraciones bruscas entre un frame y el siguiente. El sistema esta diseñado para que sea capaz de funcionar con velocidades típicas de personas humanas al andar sobre $3Km/h{-}5Km/h$, por lo que velocidades más elevadas podrían ocasionar estados de pérdida total, debido en gran parte a las limiticaciones impuestas por el modelo de movimiento escogido.

7.4.3. Correcciones de Distorsión

Debido a que el tipo de cámaras utilizadas es de gran angular, esto proporciona un amplio campo de visión. Sin embargo, este tipo de lentes conlleva una gran distorsión de la imagen. Para corregir esta distorsión se emplean unos modelos de corrección de distorsión radial y tangencial, tal y como se ha explicado en secciones anteriores.

Como se ha comentado anteriormente, la corrección de la distorsión utilizando esos modelos de corrección, **implica necesariamente una pérdida de precisión métrica**, debido al proceso de interpolación. Pero sin embargo, esta corrección es imprescindible desde el punto de vista de búsqueda de correspondencias entre un par estéreo de imágenes. Por lo tanto, se puede considerar a la distorsión como una fuente de error intrínseca al propio sistema, y el valor de este error queda determinado en gran medida por el proceso de calibración de las cámaras y los errores cometidos al calcular los parámetros de distorsión de cada una de las cámaras.

7.5. Tiempos de Cómputo

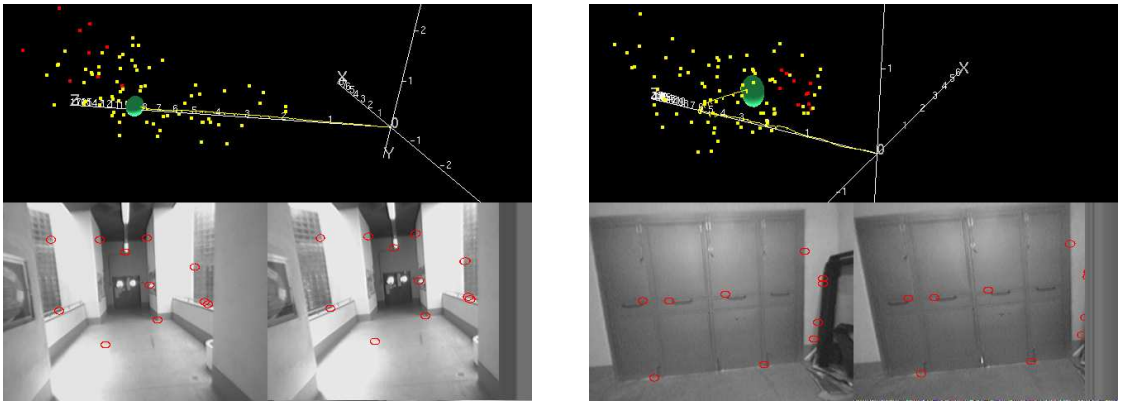
Respecto al tiempo de cómputo, la implementación en tiempo real impone una restricción de tiempo que implica no exceder de los 33 ms, considerando una velocidad de captura de 30 $frames/seg$. Estas restricciones de tiempo real, se cumplen para mapas cuyo tamaño de media no exceda las 120 marcas. Para tamaños mayores de mapa, lo que se realizará en un futuro será una estrategia de *divide y vencerás*, en la cuál se irán obteniendo pequeños submapas en tiempo real con el método propuesto, y se incluirá un alto nivel encargado del mantenimiento del mapa total que asocie en cada momento en el submapa que nos encontremos. Los tiempos medio de cómputo por ejecución obtenidos se pueden ver en la tabla 7.10:

Etapa del Algoritmo	**Tiempo (ms)**
Inicialización de Marcas (15)	20.00
Selección de Marcas	1.07
Predicción	0.47
Medidas	14.00
Actualización	4.96

Tabla 7.10: Comparativa de Estimaciones de Trayectoria Secuencia 4

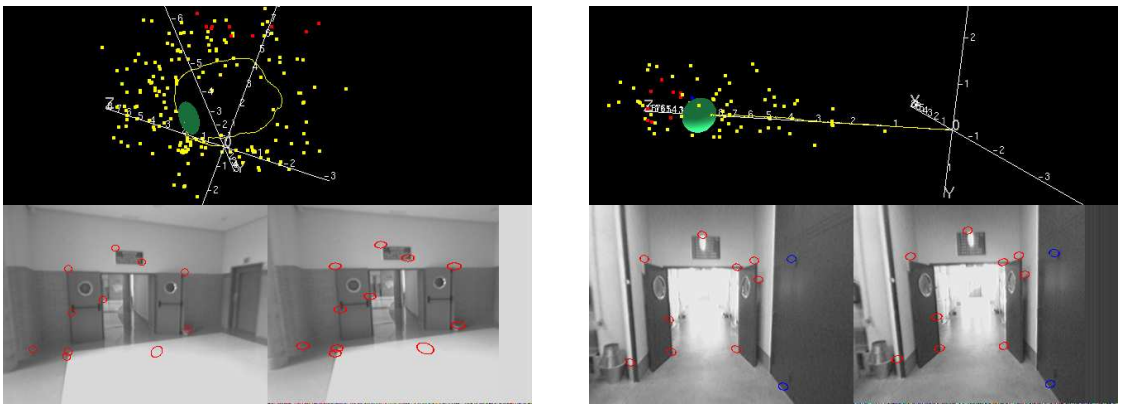
7.6. Reconstrucción de Mapas 3D

A continuación se muestran los mapas finales reconstruidos, en los que se muestra tanto la posición y orientación final de la cámara, así como la posición de las distintas marcas que forman el mapa. Los mapas reconstruidos se han obtenido con el detector de Harris, utilizando una parametrización 3D y una parametrización inversa con un umbral de profundidad de 10 m, y el método de adaptación de parches mediante homografía.



(a) Secuencia 3 (b) Secuencia 4

Figura 7.9: Reconstrucción de los Mapas 3D de las Secuencias de Test 1 y 2



(a) Secuencia 3 (b) Secuencia 4

Figura 7.10: Reconstrucción de los Mapas 3D de las Secuencias de Test 3 y 4

Capítulo 8

Conclusiones y Trabajos Futuros

En el presente trabajo se han sentado las bases de un sistema de navegación para ayuda de personas invidentes, basado en Visual SLAM a partir de la información de un sistema estéreo gran angular. Las principales conclusiones son las siguientes:

- Se obtienen buenos resultados de estimaciones de trayectorias 3D considerando entornos de interiores así como una cámara estéreo movida por la mano, mientras camina una persona a una velocidad entre 3 $Km/h - 4Km/h$, y el movimiento de la cámara sea un movimiento suave.
- El sistema es capaz de funcionar en tiempo real, siempre y cuando de manera general el tamaño del mapa no exceda de 120 marcas.
- El sistema se comporta muy bien en giros, pero sin embargo se observa cierta deriva en el eje X para trayectorias rectas.
- La calidad final del mapa 3D, depende más del modelo de medida realizado así como de la adaptación de parches, que del método de detección de características utilizado.
- Se han modelado correctamente dos tipos de parametrizaciones de marcas, así como la elección de un umbral basado en la profundidad para conmutar entre ambos tipos de parametrizaciones.
- El método de adaptación de parches basado en el cálculo de una homografía ha resultado ser el más efectivo, permitiendo que los parches se puedan medir correctamente desde más puntos de vista.

Como trabajos futuros a realizar, se proponen los siguientes:

- Realizar un SLAM de alto nivel, que permita obtener un mapa global compuesto de diversos submapas de menor tamaño. Para ello se estudiará el método basado en huellas SIFT utilizado en [39].
- Realizar un estudio más profundo en el modelo de movimiento, debido a la gran variabilidad de movimientos que puede tener una persona invidente al caminar.
- Realizar pruebas y adaptación del sistema en exteriores, así como en entornos urbanos, en donde será necesario aplicar algún criterio de descarte para aquellas marcas potencialmente no estáticas.

- Se estudiará el uso de diversos sensores como acelerómetros triaxiales (tanto para interiores como exteriores) así como GPS (para exteriores) que proporcionen una mayor robustez, así como un ground truth al sistema.
- Siempre que se desarrolla un sistema cuyo fin es servir de ayuda a personas que presentan una cierta discapacidad, es necesario obtener una realimentación por parte de los usuarios finales del sistema, sobre las necesidades de la aplicación y las características que debería tener dicho sistema. Por lo tanto, se intentará colaborar en la medida de lo posible con asociaciones de personas invidentes.

Parte III

Apéndices

Apéndice A

Cuaterniones

Los cuaterniones representan una buena herramienta para representar orientaciones en 3D. Cualquier rotación en 3D puede ser representada por una rotación simple sobre un eje determinado. En un cuaternión, el vector unitario $\mathbf{u}$ $(u_x,\ u_y,\ u_z)$ representa el eje de esta rotación y θ el ángulo de dicha rotación. Por lo tanto, la expresión del cuaternión queda representado de la siguiente manera:

$$\begin{pmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{pmatrix} = \begin{pmatrix} cos(\theta/2) \\ u_x \cdot sin(\theta/2) \\ u_y \cdot sin(\theta/2) \\ u_z \cdot sin(\theta/2) \end{pmatrix} \tag{A.1}$$

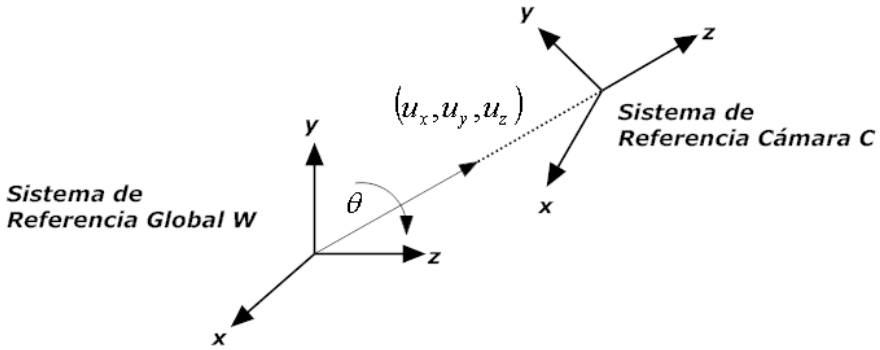


Figura A.1: Rotaciones y cuaterniones

Utilizando cuaterniones, se pueden realizar rotaciones compuestas o concatenadas con relativa facilidad. En el vector de estado utilizado en este trabajo, el cuaternión representa la rotación de la cámara con respecto al sistema de coordenadas global. A continuación se detallan las propiedades más relevantes de los cuaterniones:

- El módulo de un cuaternión, definido como la raíz cuadrada de la suma de cada uno de los elemenos al cuadrado, es siempre igual a 1.

$$q_0^2 + q_x^2 + q_y^2 + q_z^2 \ = \ 1 \tag{A.2}$$

- El conjugado de un cuaternión $\bar{q}$ representa una rotación sobre el mismo eje pero de magnitud negativa.

$$\bar{q} \;=\; \begin{pmatrix} q_0 \\ -q_x \\ -q_y \\ -q_z \end{pmatrix} \tag{A.3}$$

- La matriz de rotación R asociada con un cuaternión q queda definida como:

$$R_v \;=\; q \times v \times \bar{q} \tag{A.4}$$

donde v es un vector columna de dimensión 3×1. La expresión de la matriz de rotación R es:

$$R_{\,(3,3)} \;=\; \begin{pmatrix} q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2\,(q_x q_y - q_0 q_z) & 2\,(q_x q_z + q_0 q_y) \\ 2\,(q_x q_y + q_0 q_z) & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2\,(q_y q_z - q_0 q_x) \\ 2\,(q_x q_z - q_0 q_y) & 2\,(q_y q_z + q_0 q_x) & q_0^2 - q_x^2 - q_y^2 + q_z^2 \end{pmatrix} \tag{A.5}$$

Así, si tenemos un cuaternión q que representa la orientación 3D de una cámara y se calcula la matriz de rotación R a partir de la ecuación A.5, entonces R relaciona vectores en el sistema de referencia global W y el sistema de referencia de la cámara C y viceversa de la siguiente manera:

$$\begin{cases} v^w \;=\; R \cdot v^c = \; R^{WC} \cdot v^c \\ v^c \;=\; R^t \cdot v^w = \; R^{CW} \cdot v^w \end{cases} \tag{A.6}$$

- Composición de Rotaciones: si el cuaternión q_1 representa la rotación R^{AB} y el cuaternión q_2 representa la rotación R_{BC}, la rotación compuesta $R^{AC} \;=\; R^{AB} \cdot R^{BC}$ se representa por el producto de dos cuaterniones definido como:

$$q_3 \;=\; q_1 \times q_2 \;=\; \begin{pmatrix} q_{10}q_{20} - (q_{1x}q_{2x} + q_{1y}q_{2y} + q_{1z}q_{2z}) \\ q_{10}\begin{pmatrix} q_{2x} \\ q_{2y} \\ q_{2z} \end{pmatrix} + q_{20}\begin{pmatrix} q_{1x} \\ q_{1y} \\ q_{1z} \end{pmatrix} + \begin{pmatrix} q_{1y}q_{2z} - q_{2y}q_{1z} \\ q_{1z}q_{2x} - q_{2z}q_{1x} \\ q_{1x}q_{2y} - q_{2x}q_{1y} \end{pmatrix} \end{pmatrix} \tag{A.7}$$

La principal desventaja a la hora de representar orientaciones de esta forma, es que existe información redundante, utilizando cuatro parámetros en lugar de los tres estrictamente necesarios para representar orientaciones. Pero por otro lado, esta redundancia ayuda a tener un coste computacional mucho menor en el cálculo de rotaciones consecutivas. Un aspecto importante a tener en cuenta a la hora de trabajar con cuaterniones, es que estos deben cumplir con la propiedad de módulo unitario.

Bibliografía

[1] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping," in *Proc. of the IEEE International Conference on Robotics and Automation*, San Francisco, United States of America, Apr. 2000.

[2] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (slam): Toward exact localization without explicit localization," in *IEEE Transactions on Robotics and Automation*, vol. 17, no. 2, Apr. 2001, pp. 125–136.

[3] G. Welch and G. Bishop, "An introduction to the kalman filter," in *SIGGRAPH 2001*, University of North Carolina at Chapel Hill. Department of Computer Science, 2001.

[4] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, 2007.

[5] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and J. Nordlund, "Particle filters for positioning, navigation and tracking," in *IEEE Transactions on Signal Processing*, vol. 50, no. 2, 2002.

[6] H. Durrant-White and T. Bailey, "Simultaneous localization and mapping (slam): part 1," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 99–110, 2006.

[7] T. Bailey and H. Durrant-White, "Simultaneous localization and mapping (slam): part 2," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[8] L. M. Paz, P. Pinies, J. Neira, and J. Tardós, "6dof slam with stereo camera in hand," *IEEE Conference on Intelligent Robotics and Systems. IROS 2008.*

[9] J. Saez, F. Escolano, and A. Peñalver, "First steps towards stereo-based 6dof slam for the visually impared," *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.

[10] P. Mountney, D. Stoyanov, A. Davison, and G. Yang, "Simultaneous stereoscope localization and soft-tissue mapping for minimally invasive surgery," *MICCAI*, 2006.

[11] D. Schleicher, L. Bergasa, R. Barea, E. López, and M. Ocaña, "Real-time simultaneous localization and mapping with a wide-angle stereo camera and adaptive patches," *IEEE International Conference on Intelligent Robots and Systems*, 2006.

[12] A. J. Davison, "Mobile robot navigation using active vision," *PhD Thesis, University of Oxford*, 1998.

[13] S. Oh, S. Tariq, B. Walker, and F. Dellaert, "Laura a. clemente and a.j. davison and i. reid and j. neira and j.d. tardos," *RSS*, 2007.

[14] J. Neira and J. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.

[15] J. Shi and C. Tomasi, "Good features to track," *IEEE Proceedings on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[16] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular slam," *IEEE Transactions on Robotics*, 2008.

[17] S. Oh, S. Tariq, B. Walker, and F. Dellaert, "Map-based priors for localization," *IEEE International Conference on Intelligent Robots and Systems*, 2004.

[18] D. F. Llorca, "Procesos de calibración de cámaras: aproximación lineal y fotogramétrica," Doctorado en Electrónica, Sistemas Avanzados de Metrología de Precisión, 2004.

[19] G. Olague and R. Ramírez, "Síntesis de Imágenes a partir de Fotografías," *DYNA*, vol. 68, no. 133, pp. 17–31, Octubre 2001.

[20] G. Xu and Z. Zhang, *Epipolar Geometry in Stereo, Motion, and Object Recognition: A Unified Approach.* Kluwer Academic Publishers Norwell, MA, USA, 1996.

[21] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Prentice Hall, 2002.

[22] B. Boufama, "Reconstruction tridimensionnelle en vision par ordinateur: Cas des cameras non etalonnes," Ph.D. dissertation, INP de Grenoble, France, 1994.

[23] G. Pajares and J. de la Cruz, *Visión por Computador: Imágenes digitales y aplicaciones.* Ra-Ma, 2001.

[24] D. F. Llorca, "Sistema de detección de peatones mediante visión estereoscópica para la asistencia a la conducción," Ph.D. dissertation, Escuela Politécnica Superior, Universidad de Alcalá, 2008.

[25] M. Dhome, J. Lapresté, and J. Lavest, "Calibrage des camèras ccd," *LASMEA, Blaise Pascal University of Clermont-Ferrand*, 2003.

[26] "Documentation: Camera Calibration Toolbox for Matlab," 2007, http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

[27] D. G. Lowe, "Object recognition from local scale-invariant features," *International Conference on Computer Vision*, pp. 1150–1157, 1999.

[28] H. Bay, T. Tuytelaars, and L. Gool, "Surf: Speeded up robust features," *Proceedings of the ninth European Conference on Computer Vision*, 2006.

[29] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[30] J. Blom, "Affine invariant corner detection," Ph.D. dissertation, Utrecht University, 1991.

[31] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever, "Cartesian differential invariants in scale-space," *Journal of Mathematical Imaging and Vision*, vol. 3, pp. 327–348, November 1993.

[32] B. M. ter Haar Romeny, *Front-End Vision and Multi-Scale Image Analysis. Multi-Scale Computer Vision Theory and Applications, written in Mathematica.* Kluwer Academic Publishers, 2003.

[33] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, 1998.

[34] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, pp. 63–86, 2004.

[35] L. Paz, P. Piniés, J. Tardós, and J. Neira, "Measurement equation for inverse depth points and depth points: Large scale 6dof slam with stereo-in-hand," *Internal Document. University of Zaragoza*, December 2007.

[36] P. Zarchan and H. Musoff, "Fundamentals of kalman filtering: A practical approach," *American Institute of Aeronautics and Astronautics. Progress in Astronautics and Aeronautics*, 2000.

[37] N. Molton, A. Davison, and I. Reid, "Locally planar patch features for real-time structure from motion," *British Machine Vision Conference. BMVC 2004.*

[38] B. Liang and N. Pears, "Visual navigation using planar homographies," *IEEE International Conference on Robotics and Automation*, 2002.

[39] D. Schleicher, L. Bergasa, R. Barea, E. López, M. Ocaña, and J. Nuevo, "Real-time wide-angle stereo visual slam on large environments using sift features correction," *IEEE International Conference on Intelligent Robots and Systems*, 2007.