



Departamento de  
Informática e Ingeniería  
de Sistemas  
**Universidad Zaragoza**



# **Localización por visión omnidireccional para asistencia personal**

Daniel Gutiérrez Gómez

Directores: Alejandro Rituerto Sin y José Jesús Guerrero Campo

**Ingeniería Industrial**  
**Automatización Industrial y Robótica**

Departamento de Informática e Ingeniería de Sistemas  
Centro Politécnico Superior  
Universidad de Zaragoza

Agosto de 2011



## Localización por visión omnidireccional para asistencia personal

### RESUMEN

El problema de Localización y Reconstrucción 3D (SLAM<sup>1</sup>) plantea la posibilidad de que un móvil sea capaz de crear un mapa de un entorno desconocido y localizarse en este mapa tan solo con mediciones de sus sensores. La mayoría de implementaciones de SLAM se basan en el filtro Kalman extendido (EKF), que permite realizar predicciones del estado de un sistema a partir únicamente de mediciones del entorno.

Se pueden utilizar varios tipos de sensores para extraer la información del entorno (radar, laser, sonar, vision...). Actualmente los sensores más utilizados en SLAM son los de visión (Visual SLAM), dentro de los cuales, las cámaras omnidireccionales tienen un ángulo de visión mucho más amplio que las cámaras convencionales. A pesar de la gran ventaja que ofrece un mayor ángulo de visión en problemas como el de SLAM, existen pocas aplicaciones de Visual SLAM que utilicen cámaras omnidireccionales.

El objetivo de este proyecto es el desarrollo de una aplicación SLAM para una cámara omnidireccional a partir de una aplicación EKF-SLAM Monocular en tiempo real, programada en C++ y diseñada para su uso con cámaras convencionales. Para ello, se realizarán modificaciones sobre dos aspectos básicos: el modelo de proyección y el descriptor de puntos característicos.

El modelo de proyección debe ser adaptado a uno apropiado para sistemas omnidireccionales. Se toma el Modelo de la Esfera, un modelo no lineal que permite calcular la proyección de puntos del espacio en puntos de la imagen omnidireccional, y de puntos en la imagen en rayos tridimensionales en los que se encuentra el punto.

El descriptor de puntos característicos permite identificar los puntos en una imagen. Las imágenes omnidireccionales conllevan un modelo de proyección más complejo, así como una importante deformación y una escala variable en la imagen debido al cambio de resolución a lo largo de la dirección radial. Por lo tanto, se implementa un nuevo descriptor de puntos para cámaras omnidireccionales que los haga invariantes a cambios de escala y rotación en la imagen.

Finalmente se han llevado a cabo varios experimentos divididos en dos fases. En la primera fase se han utilizado datos facilitados por Rawseeds para hacer una comparación entre el nuevo descriptor implementado y el descriptor normal. Los resultados de estos experimentos muestran que el nuevo descriptor tiene un mejor rendimiento en el emparejamiento, que se traduce en una mejora de rendimiento en el SLAM.

En la segunda fase se ha experimentado con secuencias tomadas en el Campus Río Ebro con un casco-cámara del Grupo de Robótica, Visión y Tiempo Real, comparando los resultados obtenidos a partir de la aplicación SLAM utilizando el nuevo descriptor con los obtenidos con un dispositivo GPS también proporcionado por dicho grupo. Los resultados muestran una gran precisión en la estimación de la trayectoria en un recorrido corto, mientras que en un recorrido largo, la precisión disminuye considerablemente. Además de la deriva del error propia del algoritmo EKF-SLAM por tratarse de un algoritmo incremental, se han establecido dos posibles causas de esta disminución de la precisión, ambas consecuencia de la inobservabilidad de la escala de un SLAM monocular: por un lado, la deriva de escala a lo largo del recorrido debido a la gran longitud de este, y por otro lado, la existencia de zonas del entorno con diferentes escalas alrededor de una misma localización y que afectaría en especial a cámaras omnidireccionales por su gran ángulo de visión.

---

<sup>1</sup>del inglés Simultaneous Localization And Mapping

---



# Índice general

<b>1. Introducción</b>	<b>7</b>
<b>2. El Modelo de la Esfera</b>	<b>11</b>
<b>3. Localización y reconstrucción 3D</b>	<b>15</b>
3.1. Adquisición de datos y gestión de mapa en un SLAM Monocular . . .	18
<b>4. Adaptación de SLAM para cámaras omnidireccionales</b>	<b>21</b>
4.1. Modelo de la Esfera para EKF-SLAM . . . . .	21
4.2. Filtrado de predicciones del modelo de proyección . . . . .	22
4.3. Desarrollo de un nuevo parche invariante a escala y rotación para imágenes omnidireccionales . . . . .	24
4.3.1. Invarianza a rotación . . . . .	24
4.3.2. Invarianza a escala . . . . .	25
4.3.3. Obtención del nuevo parche para emparejamiento . . . . .	28
<b>5. Realización de experimentos</b>	<b>31</b>
5.1. Experimento 1: Evaluación del proceso de emparejamiento utilizando el nuevo parche . . . . .	31
5.1.1. Test 1: Transformación de rotación con rotación 180° de la cámara . . . . .	32
5.1.2. Test 2: Transformación de rotación con traslación de la cámara	34
5.1.3. Test 3: Factor de escala . . . . .	34
5.2. Experimento 2: Comprobación de la ejecución de un SLAM con el nuevo parche . . . . .	38
5.3. Experimento 3: SLAM con casco-cámara . . . . .	39
5.3.1. Secuencia corta . . . . .	41
5.3.2. Secuencia larga . . . . .	41
<b>6. Conclusiones</b>	<b>49</b>
<b>A. Parametrización de profundidad inversa</b>	<b>53</b>
A.1. Inicialización de puntos característicos . . . . .	54
A.2. Cambio de IDP a parametrización XYZ . . . . .	55
<b>B. La base de datos de Rawseeds</b>	<b>57</b>
<b>C. Calibración de las cámaras</b>	<b>59</b>

<b>D. Resultados completos de los tests del experimento 1</b>	<b>61</b>
<b>E. Adapting a Real-Time Monocular Visual SLAM from Conventional to Omnidirectional Cameras</b>	<b>73</b>
E.1. Introduction . . . . .	73
E.2. The Spherical Camera Model . . . . .	75
E.3. Simultaneous Localisation And Mapping . . . . .	76
E.3.1. The Spherical Camera Model for the EKF . . . . .	76
E.3.2. Data Association and Map Management . . . . .	77
E.4. New patch formulation . . . . .	77
E.4.1. Rotation invariance . . . . .	77
E.4.2. Scale invariance . . . . .	78
E.4.3. Computation of patch transformation . . . . .	79
E.5. Experiments . . . . .	80
E.5.1. Experiment 1 . . . . .	81
E.5.2. Experiment 2 . . . . .	84
E.6. Conclusion . . . . .	85

# Capítulo 1

## Introducción

El problema de localización y mapeado 3D simultáneos recibe el nombre de SLAM (Simultaneous Localization And Mapping) y ha sido ampliamente tratado en los últimos años. La resolución del SLAM permite construir un mapa de un entorno desconocido a partir de mediciones con un sensor sobre ese entorno y al mismo tiempo localizar dicho sensor en el mapa. El SLAM se formula normalmente de manera probabilística, es decir, la estimación de la posición del sensor y el mapa son calculadas como una distribución de probabilidad. Por ello, la mayoría de implementaciones se basan en una variante del filtro Kalman, conocido como filtro Kalman extendido (EKF) que linealiza las ecuaciones no lineales presentes en el modelo de proyección mediante aproximación de Taylor.

Se pueden utilizar varios tipos de sensores para extraer la información del entorno necesaria (radar, laser, sonar, vision...). En el pasado uno de los sensores más utilizados en aplicaciones SLAM fue el telémetro láser [21], pero debido a su alto coste y sus limitaciones, las investigaciones recientes se han movido hacia el uso de sensores de visión, más baratos y capaces de obtener mayor información del entorno, lo que se conoce como Visual SLAM.

En las aplicaciones de Visual SLAM, las proyecciones en la imagen de puntos característicos, también llamados características locales, se toman como medidas. Para extraer y guardar estos puntos se usan un extractor y un descriptor de características respectivamente. El extractor de características procesa la imagen y detecta los puntos claves en ella, que son aquellos en los que tiene lugar un cambio brusco de iluminación en las direcciones horizontal y vertical (esquinas). El procesamiento de la imagen es un paso que consume bastante tiempo de computación, lo cual es crítico en una aplicación en tiempo real. Centrándose en este inconveniente Rosten et. al. [24] desarrollaron el algoritmo de extracción de features FAST<sup>1</sup> y compararon su extractor FAST con otros extractores ampliamente utilizados, mostrando que el extractor FAST ofrece un mejor rendimiento tanto en coste computacional como en repetibilidad viendo la misma escena desde diferentes posiciones.

El descriptor proporciona un identificador a cada punto extraído de forma que pueda ser reconocido en futuras medidas. El descriptor más básico es un parche de cierto tamaño centrado en el punto extraído, aunque existen descriptores más sofisticados como SIFT [18], SURF [5], LBP [15], etc.

El presente proyecto está enfocado hacia Visual SLAM con sistemas de visión

---

<sup>1</sup>del inglés, Features Accelerated Segment Test

---

omnidireccionales. Dichos sistemas han experimentado un gran desarrollo en los últimos años, impulsado por la gran influencia que tiene el campo de visión en aplicaciones como localización, navegación o vigilancia. En un SLAM, gracias al campo de visión de 360° de las cámaras omnidireccionales, los puntos característicos detectados permanecen más tiempo en la imagen que en el caso de las cámaras convencionales. Este efecto se puede apreciar de forma clara bajo grandes rotaciones de la cámara donde una cámara convencional pierde gran parte de sus puntos debido a su estrecho campo de visión, mientras que la cámara omnidireccional es capaz de seguir detectando dichos puntos. El aumento de la vida de los puntos característicos en la imagen se traduce en una mejor estimación de la posición de los puntos en el mapa 3D, una menor necesidad de inicializar nuevos puntos y un incremento de la robustez del SLAM.

Dentro de las cámaras omnidireccionales existen diversos tipos según sus características constructivas:

- *Cámara rotatoria.* Se trata de una cámara convencional con un sistema mecánico, el cual le permite moverse en una trayectoria circular y capturar imágenes de todo el entorno.
- *Conjuntos de cámaras convencionales.* Cada una de las cámaras enfoca en una dirección en una configuración circular o esférica. Las imágenes recogidas se pueden unir para formar una sola imagen panorámica.
- *Sistemas dióptricos.* Estos sistemas están compuestos por cámaras convencionales y lentes de gran angular como las lentes de ojo de pez.
- *Sistemas catadióptricos.* Son la combinación de cámaras convencionales y espejos.

En aplicaciones de robótica los sistemas omnidireccionales más usados son los catadióptricos. Los sistemas catadióptricos han sido estudiados por Baker y Nayar [3] que probaron que los que incorporan espejos elípticos, hiperbólicos y parabólicos son los únicos que dan lugar a sistemas con un único centro de proyección. De estos sistemas, los más utilizados son el hiper-catadióptrico, que está compuesto por un espejo hiperbólico y una cámara perspectiva y el para-catadióptrico, que utiliza un espejo parabólico y una cámara ortográfica. El presente proyecto se centrará en el uso de sistemas de visión hipercatadióptricos. Dichos sistemas ya han sido utilizados en aplicaciones como vigilancia [28], navegación [6], localización [13], tele-presencia [13] y reconstrucción 3D [13].

Las imágenes omnidireccionales obtenidas con cámaras catadióptricas conllevan un modelo de proyección más complejo, así como una importante deformación y una escala variable en la imagen debido al cambio de resolución a lo largo de la dirección radial dependiente de la forma del espejo. Por lo tanto, es necesario diseñar un nuevo descriptor para estas cámaras. En este sentido Svoboda y Padjla [29] proponen el uso de parches con tamaño y forma variable (ventanas activas). Sus experimentos muestran que las ventanas activas proporcionan mejores resultados de emparejamiento que ventanas cuadradas. Ieng et. al. [7] proponen la computación de parches con diferentes aperturas angulares para hacer frente a los problemas de emparejamiento derivados de la resolución variable de la cámara. Scaramuzza et. al. [26] aprovechan que las líneas verticales de la escena se proyectan en líneas rectas orientadas radialmente en la imagen omnidireccional para proponer un método para extraer y emparejar líneas verticales

con descriptores invariantes a rotación y aplican este método a un EKF-SLAM. En [1], Andreasson et. al. proponen un descriptor SIFT modificado que no proporciona invarianza a escala y obtienen invarianza a rotación rotando cada parche extraído a la misma orientación global. Lu and Zheng [19] combinan el parche invariante a rotación de Andreasson con un extractor FAST y un descriptor CS-LBP y lo comparan con el algoritmo SIFT para extracción y emparejamiento.

Además, para poder extraer información sobre el entorno de la imagen, es necesario un modelo de proyección más complejo. Para sistemas catadióptricos, el más utilizado es el llamado Modelo de Proyección de la Esfera propuesto por Geyer y Daniilidis [12] y ampliado por Barreto y Araujo [4].

El objetivo de este proyecto es implementar un SLAM monocular en tiempo real con visión omnidireccional y aplicarlo a un casco con una cámara omnidireccional acoplada proporcionado por el Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza . Para ello, en primer lugar, se va a modificar el código de una aplicación desarrollada para cámaras perspectivas adaptando el modelo de proyección directo e inverso, los modelos diferenciales y el descriptor de características necesario con imágenes omnidireccionales que proporcione invarianza a rotación y escala. En segundo lugar se evaluará la nueva aplicación mediante la experimentación con secuencias tomadas por una cámara catadióptrica instalada sobre un robot obtenidas en el proyecto Rawseeds. En tercer lugar se realizará un experimento con el casco-cámara proporcionado por la Universidad de Zaragoza. Por último se pretende presentar los resultados en una conferencia internacional relevante con revisión por pares. En particular se ha enviado una contribución al 11th OMNIVIS (Anexo E) y en este momento estamos a la espera de la respuesta de los revisores.

---

## Capítulo 2

# El Modelo de la Esfera

El Modelo de la Esfera es un modelo de proyección unificado válido para todos los sistemas catadióptricos centrales. El modelo toma como origen de coordenadas  $\mathbf{O}$  de su sistema de referencia el origen del sistema central que está modelando, punto que varía en función del sistema utilizado. En el caso de estar modelando un sistema hiper-catadióptrico el punto central se encuentra en uno de los focos de la hipérbola generatriz. Si el sistema es un sistema para-catadióptrico este punto recae en el foco de la parábola, y en el caso de estar modelando una cámara perspectiva convencional coincide con el centro óptico de la cámara. En el modelo se define una esfera unitaria  $S$  centrada en el origen de coordenadas y un punto conocido como centro óptico virtual  $\mathbf{C}_P = (0, 0, -\xi)^T$ .

La información referente al espejo está encapsulada en dos parámetros característicos ( $\xi$  y  $\psi$ ). El parámetro  $\xi$ , distancia entre el origen del sistema y el centro óptico virtual, codifica el tipo de sistema que se está modelando y su geometría, siendo  $\xi = 0$  para cámaras perspectivas,  $\xi = 1$  cuando el sistema es para-catadióptrico,  $0 < \xi < 1$  para los sistemas hiper-catadióptricos y  $\xi < 0$  cuando se modela un sistema dióptrico con distorsión radial. En la Tabla 2.1 se puede ver el valor de los parámetros del modelo de la esfera para cada tipo de sistema.  $d$  representa la distancia entre focos y  $4p$  el *latus rectum*.

Tomando un punto cualquiera de la escena  $\mathbf{X}_w$ , visible por el sistema catadióptrico y expresado en coordenadas homogéneas respecto a un sistema de referencia absoluto, su proyección según el Modelo de la Esfera se puede realizar en tres pasos (figuras 2.1 y 2.2).

En primer lugar se transforma el punto  $\mathbf{X}_w$  en un rayo proyectivo  $\mathbf{x}$  en el sistema de referencia de la cámara. Esto se consigue mediante la matriz de proyección  $\mathbf{P}$

$$\mathbf{x} = \mathbf{P}\mathbf{X}_w = \mathbf{R}[\mathbf{I} - \mathbf{C}_0]\mathbf{X}_w \quad (2.1)$$

donde  $\mathbf{R}$  y  $\mathbf{C}_0$  representan la rotación y el desplazamiento del sistema de coordenadas del modelo respecto el sistema de coordenadas absoluto.

En segundo lugar, el rayo  $\mathbf{x}$  se proyecta sobre la esfera unitaria  $S$  centrada en el origen  $\mathbf{O}$ . El punto de intersección se proyecta a su vez en un plano de proyección virtual  $\pi$  a través del centro óptico virtual  $\mathbf{C}_P = (0, 0, -\xi)^T$ . Estos dos pasos se

	$\xi$	$\psi$
Espejo parabólico	1	$1 + 2p$
Espejo hiperbólico	$\frac{d}{\sqrt{d^2+4p^2}}$	$\frac{d+2p}{\sqrt{d^2+4p^2}}$
Espejo elíptico	$\frac{d}{\sqrt{d^2+4p^2}}$	$\frac{d-2p}{\sqrt{d^2+4p^2}}$
Cámara perspectiva	0	1

Tabla 2.1: Parámetros del modelo de la esfera [4]

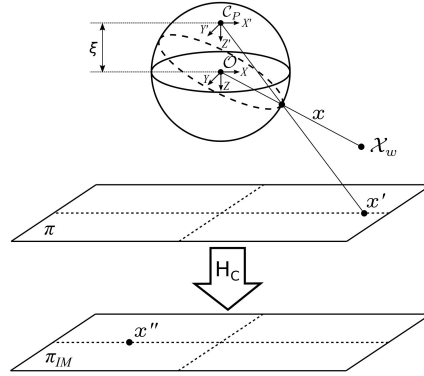


Figura 2.1: Proyección de un punto 3D  $\mathbf{X}_w$  en el plano imagen utilizando el Modelo de la Esfera

resumen en la función no lineal  $\tilde{h}$ :

$$\mathbf{x}' = \tilde{h}(\mathbf{x}) = \begin{pmatrix} x \\ y \\ z + \xi \sqrt{x^2 + y^2 + z^2} \end{pmatrix} \quad (2.2)$$

El tercer paso consiste en la proyección del plano virtual  $\pi$  al plano imagen  $\pi_{IM}$  mediante la transformación proyectiva  $H_C$ :

$$\mathbf{x}'' = H_C \mathbf{x}' \quad (2.3)$$

$$H_C = K_C R_C M_C \quad (2.4)$$

$$K_C = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$$M_C = \begin{bmatrix} \psi - \xi & 0 & 0 \\ 0 & \xi - \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -\eta & 0 & 0 \\ 0 & \eta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

donde  $K_C$  incluye los parámetros intrínsecos de la cámara perspectiva,  $M_C$  incluye los parámetros del espejo y  $R_C$  es la matriz de rotación entre la cámara y el espejo.



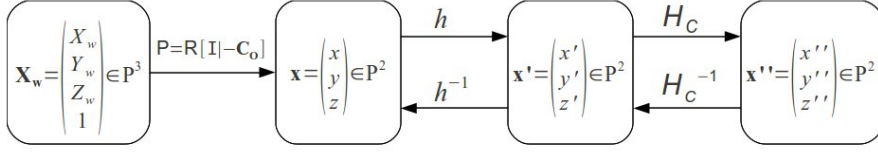


Figura 2.2: Pasos de la proyección del Modelo de la Esfera

Asumiendo que la cámara y el espejo están alineados ( $R = I$ ), la matriz de transformación  $H_C$  queda:

$$H_C = \begin{bmatrix} -\eta f_x & 0 & u_0 \\ 0 & \eta f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \gamma_x & 0 & u_0 \\ 0 & \gamma_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

donde  $\gamma_x = -\eta f_x$  y  $\gamma_y = \eta f_y$  son las distancias focales generalizadas del sistema catadióptrico completo, siendo  $\eta$  un parámetro del espejo y  $f_x$  y  $f_y$  las distancias focales de la cámara perspectiva. Hay que tener en cuenta que, dado que para un sistema catadióptrico se suelen calibrar la cámara y el espejo como un conjunto, el parámetro  $\eta$  (y por tanto, el parámetro  $\psi$ ) no se determina de forma explícita, sino que está implícito en las distancias focales generalizadas.

Finalmente, para obtener las coordenadas euclídeas del punto proyectado en la imagen, mediante la función  $\mathbf{f}_u(\mathbf{x}'')$  se divide el vector  $\mathbf{x}''$  correspondiente a las coordenadas homogéneas de la proyección entre la tercera componente:

$$\mathbf{p} = \begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{f}_u(\mathbf{x}'') = \begin{pmatrix} \frac{x''}{z''} \\ \frac{y''}{z''} \end{pmatrix} \quad (2.8)$$

El procedimiento inverso de la proyección permite obtener el rayo en el que está contenido cualquier punto de la imagen y viene definido por la inversa de la matriz de transformación  $H_C$  y por la función inversa de  $h$ . Partiendo de la proyección del punto en la imagen expresada en coordenadas homogéneas  $\mathbf{x}'' = (u, v, 1)^T$ :

$$\mathbf{x}' = H_C^{-1} \mathbf{x}'' \quad (2.9)$$

$$\mathbf{x} = h(\mathbf{x}')^{-1} = \begin{pmatrix} x' \\ y' \\ z' - \frac{\xi(x'^2 + y'^2 + z'^2)}{\xi z' + \sqrt{(1-\xi^2)(x'^2 + y'^2) + z'^2}} \end{pmatrix} \quad (2.10)$$

---

## Capítulo 3

# Localización y reconstrucción 3D

El problema de Localización y Reconstrucción 3D (SLAM<sup>1</sup>) plantea la posibilidad de que un robot (o cualquier móvil) situado en una localización desconocida en un entorno desconocido sea capaz de construir progresivamente un mapa consistente del entorno al mismo tiempo que determina su localización dentro de ese mapa. Thurn, Burgard y Fox [32] afirman que el problema de SLAM es uno de los problemas fundamentales de la robótica móvil, ya que su resolución permitiría a un robot ser completamente autónomo. En la figura 3.1 se muestra esquemáticamente el resultado de una aplicación SLAM.

El problema de SLAM se formula de manera probabilística, es decir, las estimaciones del mapa y de la posición del móvil son computadas como distribuciones de probabilidad ([16]). Por esa razón, uno de los primeros algoritmos de SLAM que aparecieron y los más usados actualmente están basados en el filtro de Kalman extendido (EKF<sup>2</sup>). El EKF es una versión del filtro de Kalman, un filtro recursivo que predice el estado de sistemas lineales mediante una distribución gaussiana multidimensional parametrizada por un vector de medias y una matriz de covarianzas. La diferencia entre el EKF y el filtro de Kalman radica en que para el EKF no es necesaria la linealidad de los sistemas, condición fundamental en el filtro de Kalman, sino que se exige tan sólo que sean derivables. De este modo las funciones de cambio de estado o de medida no lineales son aproximadas por una serie de Taylor de primer orden, de forma que sea posible calcular las covarianzas del sistema.

En una aplicación EKF-SLAM el estado  $\mathbf{x}_k$  del sistema completo en un instante  $k$  viene dado por el estado del móvil  $\mathbf{x}_{v,k}$  y por las localizaciones  $\mathbf{y}$  de los puntos del mapa 3D :

$$\mathbf{x} = \underbrace{(\mathbf{r}_k, \mathbf{q}_k, \mathbf{V}_k, \omega_k)}_{\mathbf{x}_{v,k}}, \underbrace{(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m)}_{\mathbf{y}} \quad (3.1)$$

donde  $\mathbf{r}_k(3 \times 1)$  es la posición del móvil,  $\mathbf{q}_k(4 \times 1)$  es el cuaternio de su orientación, y  $\mathbf{V}_k(3 \times 1)$  y  $\omega_k(3 \times 1)$  son sus velocidades lineal y angular respectivamente. Las posiciones de los puntos del mapa no dependen del instante de tiempo ya que se consideran puntos fijos.

---

<sup>1</sup>del inglés Simultaneous Localization And Mapping

<sup>2</sup>del inglés Extended Kalman Filter

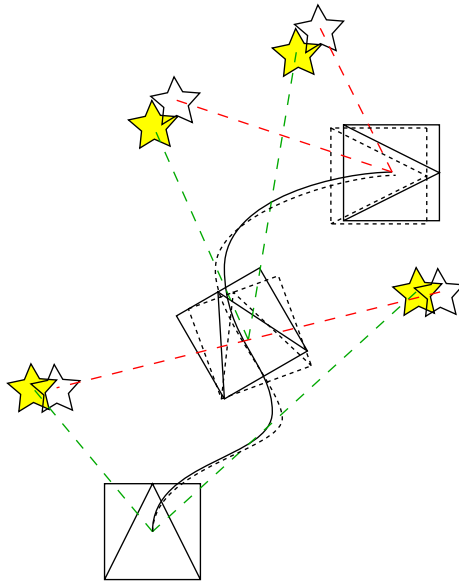


Figura 3.1: Aplicación de SLAM. En línea continua se representa la trayectoria y las posiciones reales del robot, y en línea discontinua las estimaciones del SLAM. Las mediciones del entorno se representan como estrellas, amarillas con línea discontinua verde la primera vez que son observadas y blancas con línea roja discontinua cuando se reconocen desde otra posición.

Para poder llevar a cabo la estimación de un nuevo estado en el instante  $k$  es necesario definir previamente un modelo de movimiento y un modelo de observación.

El modelo de movimiento se describe mediante la siguiente función:

$$\mathbf{x}_{\mathbf{v},k} = \mathbf{f}_{\mathbf{v}}(\mathbf{x}_{\mathbf{v},k-1}, \mathbf{u}_k) + \mathbf{w}_k \quad (3.2)$$

donde  $\mathbf{f}_{\mathbf{v}}(\cdot)$  es la función que modela el movimiento del móvil,  $\mathbf{x}_{\mathbf{v},k-1}$  es el estado anterior del móvil,  $\mathbf{u}_k$  es la entrada de control independiente de las observaciones y del mapa y  $\mathbf{w}_k$  son perturbaciones aditivas con distribución gaussiana, no correlacionadas, de media cero y con covarianza  $\mathbf{Q}_k$ . En este proyecto se va a utilizar el modelo de velocidad constante y aceleraciones lineal y angular desconocidas. Dichas aceleraciones son tratadas como ruido gaussiano y, por lo tanto, englobadas dentro del término  $\mathbf{w}_k$  de la ecuación.

Por otro lado, el modelo de medida se describe de la siguiente forma:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad (3.3)$$

donde  $\mathbf{h}(\cdot)$  describe el modelo de proyección,  $\mathbf{x}_k$  es el estado total y  $\mathbf{v}_k$  son errores de observación aditivos, con distribución gaussiana, no correlacionadas, de media cero y con covarianza  $\mathbf{R}_k$

En cada iteración del EKF la estimación se realiza en dos pasos. El primer paso consiste en la predicción del estado y de la covarianza de dicho estado en función de la información del paso anterior. El segundo paso es el de la actualización del estado con la información obtenida de las medidas.

■ **Predicción**

La predicción del nuevo estado se realiza mediante una función de transición de estado. La transición de estado se toma como un proceso de Markov en el que el estado  $\mathbf{x}_k$  depende únicamente del estado anterior  $\mathbf{x}_{k-1}$  y de la entrada de control  $\mathbf{u}_k$  y es independiente de las observaciones y del mapa. De este modo las ecuaciones para la predicción del nuevo estado y la covarianza quedan:

*Predicción del estado*

$$\mathbf{x}_{k|k-1} = f(\mathbf{x}_{k-1|k-1}, \mathbf{u}_k) \quad (3.4)$$

*Predicción de la covarianza*

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1} \mathbf{P}_{k-1|k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_k \quad (3.5)$$

donde  $f$  es la función de cambio de estado, que actualiza el estado del móvil según el modelo de movimiento ( $\mathbf{x}_{v,k|k-1} = \mathbf{f}_v(\mathbf{x}_{v,k-1|k-1}, \mathbf{u}_k)$ ) y no produce ningún cambio en el estado de los puntos del mapa ( $\mathbf{y}_{k|k-1} = \mathbf{y}_{k-1|k-1}$ ), ya que se asume que son puntos fijos.  $\mathbf{x}_{k|k-1}$  y  $\mathbf{P}_{k|k-1}$  son el estado y la covarianza de ese estado estimados en el momento  $k$  a partir de las mediciones hechas hasta el momento anterior  $k-1$ .  $\mathbf{F}_{k-1}$  es el jacobiano de la función de cambio de estado.

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

y  $\mathbf{Q}_k$  es la matriz de covarianzas del ruido del sistema.

■ **Actualización**

Tras la primera predicción del estado del sistema mediante el modelo de movimiento se realiza una actualización de esta predicción basada en las medidas de los sensores en el momento actual  $k$ . Para ello, en primer lugar se calcula la innovación de la medida  $\nu_k$  que representa la diferencia entre la medida real  $\mathbf{z}_k$  realizada por los sensores y la predicción de la medida  $h(\mathbf{x}_{k|k-1})$  dada por el modelo de observación. Esta innovación tiene asociada una covarianza  $\mathbf{S}_k$  en la que se tiene en cuenta los posibles errores de medida.

*Innovación de la medida*

$$\nu_k = \mathbf{z}_k - h(\mathbf{x}_{k|k-1}) \quad (3.6)$$

*Covarianza de la innovación*

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \quad (3.7)$$

donde  $\mathbf{H}_k$  es el jacobiano de la función de medida.

$$\mathbf{H}_k = \left. \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}_{k|k-1}}$$

A continuación se calcula la ganancia de Kalman  $W_k$ , que señala cuanto se puede confiar en las mediciones a la hora de corregir la predicción a priori. En función de este valor se le da más o menos peso a la innovación de la medida al calcular el estado y su covarianza finales en el instante  $k$ .

*Ganancia de Kalman*

$$W_k = P_{k|k-1} H_k^T S_k^{-1} \quad (3.8)$$

*Actualización del estado*

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + W_k \nu_k \quad (3.9)$$

*Actualización de la covarianza*

$$P_{k|k} = P_{k|k-1} - W_k S_k W_k^T \quad (3.10)$$

### 3.1. Adquisición de datos y gestión de mapa en un SLAM Monocular

Previamente se ha definido de forma genérica el algoritmo de una solución para SLAM basada en EKF para cualquier tipo de sensor. Sin embargo, el presente proyecto se basa en un SLAM con un solo sensor de visión (SLAM Monocular), por lo que a continuación se hará hincapié en sus particularidades.

El uso de sensores de visión para SLAM viene motivado sobre todo por su bajo coste y por su capacidad de proporcionar gran cantidad de información del entorno. Sin embargo, presentan el inconveniente de que es necesario un mayor procesamiento de los datos, en este caso imágenes. Las medidas en SLAM con sensores de visión se corresponden con las proyecciones en la imagen de puntos relevantes del entorno (características locales). Para detectar estos puntos se utilizan extractores de características que consisten en algoritmos que procesan la imagen buscando los puntos en los que tiene lugar un cambio brusco de iluminación en las direcciones horizontal y vertical. El procesamiento de la imagen es un paso que consume bastante tiempo de computación, lo cual es crítico en una aplicación en tiempo real. Por ello, en la aplicación en la que se basa el presente proyecto se utiliza el extractor FAST<sup>3</sup>. Dicho extractor fue desarrollado por Rosten et. al. ([24]) y se ha comparado con otros extractores ampliamente utilizados. Los resultados muestran que el extractor FAST ofrece un mejor rendimiento tanto en coste computacional como en repetibilidad viendo la misma escena desde diferentes posiciones.

Por otro lado, para poder hacer un seguimiento de los puntos extraídos, es necesario proporcionarles un identificador o descriptor para que puedan ser reconocidos en las siguientes imágenes. En este proyecto usaremos un identificador sencillo consistente en un parche cuadrado centrado en el punto característico, frente a otros descriptores más sofisticados como el SIFT. La razón es que, pese a que el descriptor SIFT es invariante a cambios de escala y orientación en imágenes convencionales, viene ligado a la utilización de un algoritmo de extracción de puntos (DoG) que es mucho más complejo computacionalmente que el extractor FAST. Además, como se propone en este proyecto de fin de carrera es posible hacer frente a estos cambios de escala y de orientación de los puntos característicos en imágenes omnidireccionales, mediante la aplicación de transformaciones afines.

Una vez presentado el algoritmo EKF-SLAM básico y definido el proceso de medición a partir de una imagen, se explicará más detalladamente el proceso de asociación de datos y gestión de puntos característicos en el SLAM monocular, que se resume de forma esquemática en el algoritmo 1 al final del capítulo.

<sup>3</sup>Features Accelerated Segment Test

Partiendo del estado anterior, definido por el vector de estados y la matriz de covarianzas, se calcula en primer lugar la predicción del nuevo estado mediante las ecuaciones 3.4 y 3.5. A continuación se lleva a cabo la predicción de las medidas de los puntos del mapa mediante el modelo de medida (Ec. 3.3), que en el caso de sensores de visión es un modelo de proyección, y se calcula su correspondiente covarianza (Ec. 3.7). La covarianza de la innovación define una elipse de incertidumbre para cada punto característico dentro de la cual debería encontrarse la proyección real del punto característico en la nueva imagen. Para obtener una medición de dicho punto se realiza una búsqueda activa dentro de la región de incertidumbre, en la que se selecciona como medición (o emparejamiento) el píxel con mejor correlación con el parche del punto característico. Para detectar y descartar mediciones erróneas (es decir, aquellas que no se corresponden con el punto buscado) se realiza un filtrado RANSAC de un punto [11]. Finalmente, teniendo en cuenta sólo los emparejamientos correctos, se calcula la innovación y la ganancia Kalman y se ejecuta el paso de actualización para calcular el nuevo estado.

Para la inicialización de nuevos puntos se procesa la imagen con el extractor FAST y se inicializan aquellas que tienen una mayor puntuación y que estén lo suficientemente alejados del resto de puntos del mapa.

En el caso de sensores de visión existe un problema en la inicialización de puntos debido a que la profundidad de un punto en el espacio tridimensional no es observable en la imagen, por lo que tiene que ser inicializada a un valor arbitrario con una incertidumbre muy elevada. Por esta razón, el estado de los puntos recién inicializados se expresa mediante parametrización de profundidad inversa (IDP<sup>4</sup>) que se explica detalladamente en el Anexo A. Dicha representación consta de 6 parámetros: las coordenadas de la cámara  $(x_i, y_i, z_i)$  en el momento de la inicialización del punto, el ángulo de elevación  $\theta_i$ , el ángulo de azimuth  $\phi_i$  y el inverso de la profundidad  $\rho_i$ . Los puntos son inicializados con una estimación arbitraria de  $\rho_{0i}$  para la profundidad inversa. En sucesivas observaciones del punto, la estimación de la profundidad se va afinando gradualmente hasta que, si su incertidumbre decrece por debajo de cierto umbral, el estado del punto pasa a representarse por sus coordenadas cartesianas en la referencia absoluta.

Nótese que la inobservabilidad de la profundidad en la imagen, tiene el problema asociado de la inobservabilidad de la escala en el SLAM Monocular, dependiendo ésta del valor arbitrario que se le da a la estimación inicial de la profundidad. Esto implica que una implementación de SLAM monocular requiere de un método alternativo para determinar la escala del mapa.

En lo que respecta a la ejecución en tiempo real, el coste computacional de la actualización del estado crece de forma cuadrática con el número de puntos en el mapa, superando el límite impuesto por la velocidad de adquisición de imágenes cuando el mapa es demasiado grande. Por lo tanto aquellos puntos que no son emparejados o son emparejados erróneamente tras varios intentos se marginalizan fuera del vector de estados, guardándolos en otro vector independiente del SLAM en el primer caso y borrándolos en el segundo.

---

<sup>4</sup>del inglés, Inverse Depth Parametrisation

### 3.1. ADQUISICIÓN DE DATOS Y GESTIÓN DE MAPA EN UN SLAM MONOCULAR

---

---

**Algorithm 1** Algoritmo EKF-SLAM para sistemas de visión

---

**Require:**  $\mathbf{x}_{k-1|k-1}$ ,  $\mathbf{P}_{k-1|k-1}$ ,  $\mathbf{u}_k$ , *im\_actual*, *parches\_puntos*

**Ensure:**  $\mathbf{x}_{k|k}$ ,  $\mathbf{P}_{k|k}$ , *parches\_puntos\_nuevos*

$(\mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}) = \text{Prediccion Movimiento}(\mathbf{x}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}, \mathbf{u}_k, \mathbf{Q}_k)$

**if** Puntos en el vector de estado  $\neq 0$  **then**

$(\mathbf{h}_k, \mathbf{S}_k) = \text{Prediccion Medida}(\mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}, \mathbf{R}_k)$

$\mathbf{z}_k = \text{Busqueda y Emparejamiento}(\mathbf{h}_k, \mathbf{S}_k, \textit{im\_actual}, \textit{parches\_puntos})$

$(\mathbf{x}_{k|k}, \mathbf{P}_{k|k}) = \text{Filtrado y Actualizacion}(\mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}, \mathbf{z}_k, \mathbf{h}_k, \mathbf{S}_k, \mathbf{R}_k)$

**end if**

$(\mathbf{z}_{init}, \textit{parches\_puntos\_nuevos}) = \text{Extraer y Seleccionar Puntos}(\textit{im\_actual})$

$(\mathbf{y}_{init}, \mathbf{P}_y) = \text{Inicializar Puntos}(\mathbf{z}_{init}, \mathbf{R}_k)$

$\mathbf{x}_{k|k} = [\mathbf{x}_{k|k} \ \mathbf{y}_{init}]$

$\mathbf{P}_{k|k} = [\mathbf{P}_{k|k} \ \mathbf{P}_y]$

---



## Capítulo 4

# Adaptación de SLAM para cámaras omnidireccionales

En el capítulo anterior se ha presentado el algoritmo para SLAM con EKF y se ha profundizado en las particularidades de una aplicación de visual SLAM monocular.

Dado que partimos de una aplicación ya desarrollada para cámaras convencionales ([8]), en este capítulo nos centraremos en introducir y desarrollar las modificaciones necesarias sobre esta aplicación para que sea posible su uso con cámaras catadióptricas. Dichas modificaciones se evalúan mediante los experimentos descritos en el siguiente capítulo. El uso de sistemas de visión omnidireccionales en lugar de una cámara convencional conlleva, además de la modificación del modelo de proyección por el Modelo de la Esfera, otra serie de cambios en los algoritmos de gestión de puntos característicos debido, por un lado, al propio modelo de proyección, y por otro, a la gran distorsión, deformación y escala variable que presenta una imagen omnidireccional frente a una imagen normal.

### 4.1. Modelo de la Esfera para EKF-SLAM

En un SLAM para sistemas catadióptricos el modelo de medida viene dado por el Modelo de la Esfera, el cual incluye funciones no lineales. Por lo tanto, es necesaria su linealización obteniendo una expresión del Jacobiano que permite calcular la ganancia Kalman ([23]). Usando la misma notación que en el Capítulo 2, el Jacobiano correspondiente al Modelo de la Esfera se obtiene de la siguiente forma:

$$J_{ME} = \frac{\partial \mathbf{u}}{\partial \mathbf{x}} = \underbrace{\frac{\partial \mathbf{u}}{\partial \mathbf{x}''}}_{J_{fu}} \underbrace{\frac{\partial \mathbf{x}''}{\partial \mathbf{x}'}}_{H_C} \underbrace{\frac{\partial \mathbf{x}'}{\partial \mathbf{x}}}_{J_h} \quad (4.1)$$

con

$$J_{fu} = \begin{bmatrix} \frac{1}{z''} & 0 & -\frac{x''}{z''^2} \\ 0 & \frac{1}{z''} & -\frac{y''}{z''^2} \end{bmatrix} \quad (4.2)$$

$$J_h = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{\xi x}{\sqrt{x^2+y^2+z^2}} & \frac{\xi y}{\sqrt{x^2+y^2+z^2}} & 1 + \frac{\xi z}{\sqrt{x^2+y^2+z^2}} \end{bmatrix} \quad (4.3)$$

También es necesario obtener el Jacobiano de la proyección inversa, que permite calcular la propagación de incertidumbres desde la imagen hasta la escena y por lo tanto, hacer una primera estimación de la covarianza del estado de un punto recién inicializado. Utilizando la notación del Capítulo 2, el Jacobiano de la función de proyección inversa se puede calcular como el producto de los jacobianos correspondientes a cada paso de la función.

$$J_{ME}^{-1} = \frac{\partial \mathbf{x}}{\partial \mathbf{x}''} = \frac{\partial \mathbf{x}}{\partial \mathbf{x}'} \frac{\partial \mathbf{x}'}{\partial \mathbf{x}''} = J_{\tilde{h}^{-1}} H_C^{-1} \quad (4.4)$$

siendo  $J_{\tilde{h}^{-1}}$  el jacobiano de la inversa de la función no lineal  $\tilde{h}$ :

$$J_{\tilde{h}^{-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{\partial z}{\partial x'} & \frac{\partial z}{\partial y'} & \frac{\partial z}{\partial z'} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{\xi x'}{\chi} & -\frac{\xi y'}{\chi} & 1 - \frac{\xi(z' - \xi \frac{x'^2 + y'^2 + z'^2}{\xi z' + \chi})}{\chi} \end{bmatrix} \quad (4.5)$$

donde  $\chi = \sqrt{(1 - \xi^2)(x'^2 + y'^2 + z'^2)}$

## 4.2. Filtrado de predicciones del modelo de proyección

Como su nombre indica, un modelo de proyección no deja de ser una idealización de la realidad, por lo que muchas de las medidas predichas carecerán de sentido físico al no ser visibles en la práctica por el sistema de visión. Para descartar las predicciones físicamente incoherentes se realiza un filtrado en tres fases correspondiendo a las tres posibles causas de incoherencia en la predicción.

- Puntos no visibles según el modelo de proyección teórico

Son aquellos puntos para los que el rayo  $\mathbf{x}'$  obtenido mediante la función  $\tilde{h}$  tiene componente  $z' \leq 0$ . Geométricamente son los puntos dentro del cono definido por la ecuación  $x^2 + y^2 = \frac{1-\xi^2}{\xi^2} z^2$  con  $z < 0$  tomando como origen de coordenadas el punto  $\mathbf{O}$  (Fig. 4.1).

- Puntos fuera de la imagen

Son los puntos que son proyectados en el plano imagen según el modelo de proyección, pero cuyas coordenadas quedan fuera de los límites de la imagen capturada.

- Puntos proyectados en zonas ciegas de la imagen

Por razones constructivas, las cámaras catadióptricas presentan zonas en el centro y en la periferia de la imagen que permanecen constantemente inmóviles, al corresponder a la proyección de elementos solidarios a la cámara. Para filtrar las medidas predichas en estas zonas, se enmascaran los píxeles de la imagen que se encuentren a una distancia menor de un radio mínimo o mayor de un radio máximo del punto principal. Puede darse el caso de que exista también algún elemento de sujeción de la cámara como una barra vertical que atraviesa la imagen radialmente (Fig. 4.2) que también sería necesario enmascarar.

CAPÍTULO 4. ADAPTACIÓN DE SLAM PARA CÁMARAS  
OMNIDIRECCIONALES

---

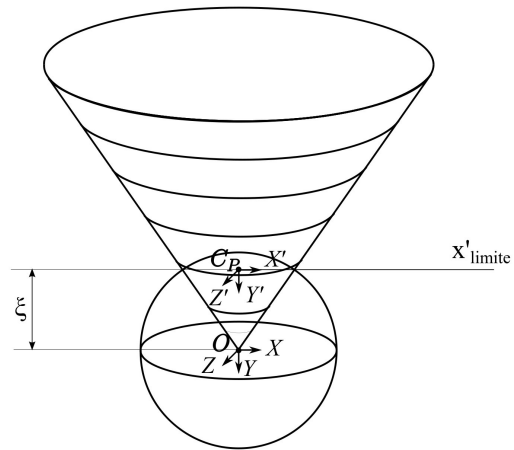


Figura 4.1: Los puntos de la escena que estén contenidos dentro del cono de la figura no se pueden proyectar en el plano imagen.

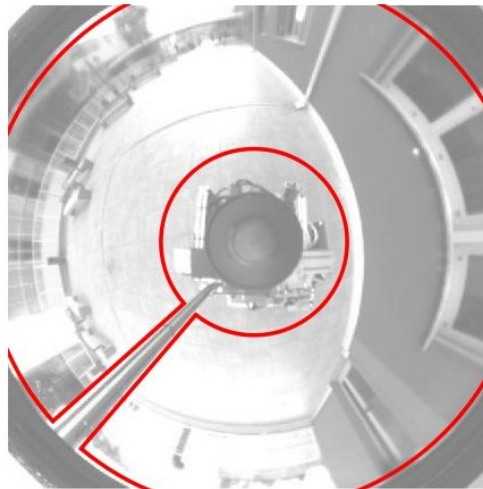


Figura 4.2: Máscara que se aplica a las cámaras omnidireccionales para no inicializar ni emparejar puntos que permanecen inmóviles en la imagen por ser solidarios a la cámara.

### 4.3. Desarrollo de un nuevo parche invariante a escala y rotación para imágenes omnidireccionales

Durante el proceso de medición de puntos característicos, además de predecir su proyección, es deseable también predecir como cambia la apariencia del parche en la imagen conforme la cámara se va desplazando para aumentar las posibilidades de conseguir emparejamientos exitosos.

En la aplicación original esto se consigue guardando un parche de tamaño mayor además del parche usado como descriptor durante la inicialización. Antes de hacer el emparejamiento de puntos, se determina una transformación homográfica [14] a partir de la variación de la posición y orientación del robot entre el momento actual y el de la inicialización del punto que se va a emparejar. Para poder construir esta transformación es necesario asumir que todos los puntos del parche se encuentran en el mismo plano de la escena 3D y que este plano es paralelo al plano imagen, algo que no puede asumirse en imágenes catadióptricas omnidireccionales.

En la aplicación modificada vamos a definir una nueva transformación para predecir la variación en la apariencia de los parches en una cámara catadióptrica. Con dicha transformación se pretende proporcionar cierta robustez respecto a los cambios de orientación y de escala del parche en la imagen. A diferencia de la transformación implementada para cámaras convencionales, la nueva transformación es más sencilla y no proporciona una predicción del cambio de perspectiva, pero presenta por un lado la ventaja de que no requiere realizar ninguna hipótesis sobre el parche en la escena 3D, y por otro lado, que, exceptuando la contribución de la profundidad en la escala del parche, las transformaciones se calculan a partir de información presente solamente en la imagen, por lo que no se requiere información sobre las variables de estado del SLAM.

#### 4.3.1. Invarianza a rotación

Para conseguir robustez respecto de la rotación de la cámara en torno al eje de revolución del espejo, en principio hemos adoptado la idea propuesta por Andreasson et al. [1], consistente en tomar mediante interpolación bilineal un parche orientado en la dirección radial desde el punto principal como descriptor y rotarlo a una orientación fija (Fig. 4.3) Sin embargo, a la hora de hacer el emparejamiento, esta solución tiene el inconveniente de que conlleva un coste computacional elevado, ya que cada parche candidato dentro de la región de búsqueda debe ser rotado también por interpolación bilineal.

Por ello, se ha combinado esta idea con la implementación de la aplicación original. En vez de extraer un parche en la dirección radial, se extrae un parche de mayor tamaño (p.ej. el doble de grande) que el descriptor en las direcciones horizontal y vertical durante la inicialización de un nuevo punto del mapa. Antes del proceso de emparejamiento, se le aplica a este parche una transformación de rotación  $R_{\Delta\theta}$  a partir de la variación del ángulo polar del punto en la imagen desde su inicialización ( $\Delta\theta = \theta - \theta_{ini}$ )

$$R_{\Delta\theta} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) & 0 \\ \sin(\Delta\theta) & \cos(\Delta\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

y a continuación se extraería el nuevo parche descriptor del centro del parche

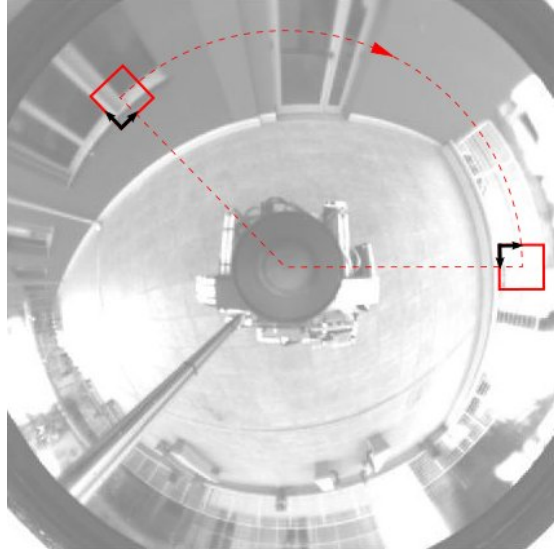


Figura 4.3: El parche de un punto característico recién inicializado, se rota a una orientación fija antes de ser guardado como descriptor para futuros emparejamientos. De esta forma el parche descriptor es invariante frente a rotación de la cámara.

transformado. De esta manera la correlación de los puntos dentro de la región de búsqueda se determina siempre en las direcciones horizontal y vertical de la imagen y la interpolación bilineal sólo se realiza una vez durante la transformación del parche grande.

### 4.3.2. Invarianza a escala

Para que el proceso de emparejamiento de un punto sea invariante a la escala, se va a formular el cambio de escala con el que un mismo punto se puede observar en distintas imágenes. Para considerar la resolución variable en la imagen catadióptrica en el cálculo del factor de escala, se ha obtenido una fórmula matemática en función de los parámetros del espejo y la posición en la imagen.

Para ello, en primer lugar se ha definido un punto 3D en coordenadas homogéneas a una distancia o profundidad  $D$  de la cámara con azimut  $\phi$  y elevación  $\theta$ . Debido a la simetría de revolución del espejo hiperbólico y por simplicidad, se toma un azimut de  $\phi = 0$  sin pérdida de generalidad. Por lo tanto, las coordenadas del punto 3D quedan  $\mathbf{X}_0 = (D \cos \theta, 0, D \sin \theta, 1)^\top$ .

Proyectando este punto en el plano imagen según el Modelo de la Esfera obtenemos el punto  $\mathbf{p}_0 = (\frac{\gamma \cos \theta}{\xi + \sin \theta}, 0, 1)^\top$ . Para expresar las coordenadas en la imagen respecto del sistema de referencia en el punto principal hemos tomado  $u_0 = v_0 = 0$  en la matriz  $H_C$ , lo que se mantendrá para el resto de la formulación. La norma del punto proyectado es la distancia desde el punto principal  $R_{im}$

$$R_{im} = \|\mathbf{p}_0\| = \frac{\gamma \cos \theta}{\xi + \sin \theta} \quad (4.7)$$

Ahora tomemos una esfera de radio  $r$  centrada en el punto 3D  $\mathbf{X}_0$  que hemos proyectado previamente. Si asumimos que  $r \ll D$ , los puntos en la superficie de la

### 4.3. DESARROLLO DE UN NUEVO PARCHE INVARIANTE A ESCALA Y ROTACIÓN PARA IMÁGENES OMNIDIRECCIONALES

esfera se pueden proyectar por medio del Jacobiano del modelo de proyección evaluado en su centro:

$$J_{p=p_0} = J_{fu} H_c J_h P \quad (4.8)$$

Después de algunos cálculos y de manipulaciones algebraicas obtenemos la siguiente expresión para el Jacobiano:

$$J_{p=p_0} = \frac{\gamma}{D(\xi + S_\theta)^2} \begin{bmatrix} S_\theta(1 + \xi S_\theta) & 0 & -C_\theta(1 + \xi S_\theta) \\ 0 & \xi + S_\theta & 0 \end{bmatrix} \quad (4.9)$$

donde  $S_\theta$  y  $C_\theta$  son  $\sin \theta$  y  $\cos \theta$  respectivamente

Y proyectando la esfera obtenemos una elipse en la imagen cuyos semieje mayor ( $r_{im}^+$ ) y semieje menor ( $r_{im}^-$ ) quedan:

$$r_{im}^+ = \gamma \frac{r}{D} \frac{1 + \xi \sin \theta}{(\xi + \sin \theta)^2} \quad (4.10)$$

$$r_{im}^- = \gamma \frac{r}{D} \frac{\xi + \sin \theta}{(\xi + \sin \theta)^2} \quad (4.11)$$

Si resolvemos (4.7) para  $\theta$  podemos poner  $\sin \theta$  como función de  $R_{im}$  en las expresiones anteriores. Después de algunas sustituciones y manipulaciones resulta una ecuación de segundo grado con  $\sin \theta$  como incógnita. Resolviendo la ecuación y seleccionando la solución con significado físico obtenemos la siguiente expresión:

$$\sin \theta = f\left(\xi, \frac{R_{im}}{\gamma}\right) = \frac{\sqrt{1 + \left(\frac{R_{im}}{\gamma}\right)^2 (1 - \xi^2)} - \xi \left(\frac{R_{im}}{\gamma}\right)^2}{1 + \left(\frac{R_{im}}{\gamma}\right)^2} \quad (4.12)$$

y sustituyendo en (4.10) y (4.11) :

$$r_{im}^+ = \gamma \frac{r}{D} \frac{1 + \xi f\left(\xi, \frac{R_{im}}{\gamma}\right)}{\left(\xi + f\left(\xi, \frac{R_{im}}{\gamma}\right)\right)^2} \quad (4.13)$$

$$r_{im}^- = \gamma \frac{r}{D} \frac{1}{\xi + f\left(\xi, \frac{R_{im}}{\gamma}\right)} \quad (4.14)$$

Según estas fórmulas el tamaño de un objeto en la imagen depende de los siguientes parámetros:

- Tamaño real del objeto ( $r$ )
- Distancia del objeto a la cámara ( $D$ )
- Parámetros del sistema cámara-espejo  $\xi$  and  $\gamma$
- Distancia de la proyección del objeto en la imagen al punto principal ( $R_{im}$ )

Nótese que el tamaño real del objeto no es relevante a la hora de calcular el cambio de escala, ya que permanece constante entre dos imágenes.

En lo que respecta a la contribución de la distancia al centro de la imagen  $R_{im}$ , debemos elegir entre las dos expresiones obtenidas anteriormente dependiendo de que semieje de la elipse proyectada anteriormente usemos para calcular el tamaño. Hay que notar que por simplicidad la distorsión radial no se ha tenido en cuenta en el modelo

## CAPÍTULO 4. ADAPTACIÓN DE SLAM PARA CÁMARAS OMNIDIRECCIONALES

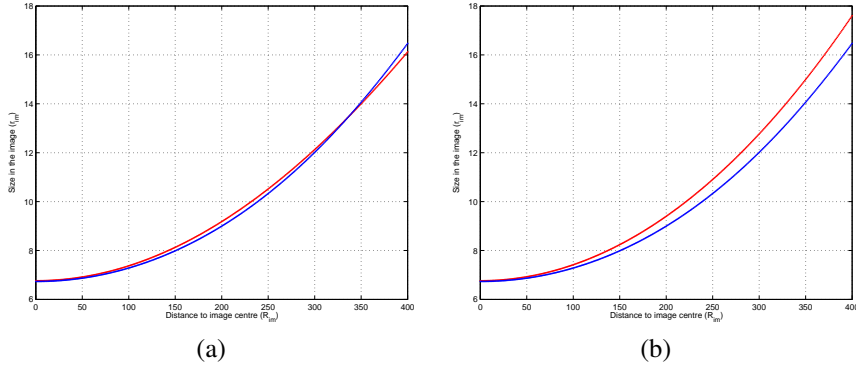


Figura 4.4: Comparación de las fórmulas teóricas para calcular los semiejes de la elipse en la que la esfera es proyectada (rojo) con las gráficas obtenidas de una simulación con el modelo de una cámara con distorsión radial (azul). La gráfica (a) corresponde al semieje menor y la gráfica (b) al semieje mayor.

de proyección. Esto, unido a la simetría de revolución del espejo, haría que la fórmula para el semieje mayor, que tiene dirección radial, fuera menos precisa que la del semieje menor.

Para confirmarlo se ha realizado una simulación usando un modelo de cámara catadióptrica con distorsión, tomando los parámetros de la cámara de los datos de calibración del casco-cámara (Anexo C) y proyectando un conjunto de esferas de radio  $r = 0,1 \text{ m}$ , a una profundidad  $D = 5 \text{ m}$  constante y con diferentes ángulos de elevación  $\theta$ . De esta manera se ha obtenido de forma experimental para cada semieje una función que relaciona su longitud con la distancia del centro de la elipse al punto principal, que se compara con la teórica (Fig. 4.4). Para realizar la elección en base a criterios cuantitativos se ha obtenido para los dos semiejes el error absoluto y el error relativo entre la función obtenida por simulación y la obtenida mediante la formulación desarrollada en esta sección (Fig. 4.5). Como se puede ver en la figura 4.5, usando el semieje mayor para calcular el factor de escala se estaría cometiendo un error absoluto máximo de más de 1 píxel y un error relativo máximo del 7 %, mientras que con la fórmula para el semieje menor el error absoluto máximo es de 0,2 píxeles y el error relativo máximo es del 2 %. Por lo tanto, se concluye que la fórmula para el semieje menor es la más apropiada para calcular el factor de escala.

De esta manera, la expresión final para calcular el factor de escala queda:

$$k = \frac{r_{im2}^-}{r_{im1}^-} = \frac{D_1 \xi + f(\xi, \frac{R_{im1}}{\gamma})}{D_2 \xi + f(\xi, \frac{R_{im2}}{\gamma})} \quad (4.15)$$

con

$$f(\xi, \frac{R_{im_i}}{\gamma}) = \frac{\sqrt{1 + (\frac{R_{im_i}}{\gamma})^2(1 - \xi^2)} - \xi(\frac{R_{im_i}}{\gamma})}{1 + (\frac{R_{im_i}}{\gamma})^2} \quad (4.16)$$

La profundidad del objeto en la escena  $D$  es la contribución más problemática ya que no es observable en la imagen. Como ya se explicó en el apartado 3,1, durante la ejecución del SLAM un punto que es detectado por primera vez es inicializado en parametrización inversa (IDP) con un valor arbitrario para la profundidad y con una incertidumbre muy elevada, por lo que no es un valor fiable para calcular el factor de

### 4.3. DESARROLLO DE UN NUEVO PARCHE INVARIANTE A ESCALA Y ROTACIÓN PARA IMÁGENES OMNIDIRECCIONALES

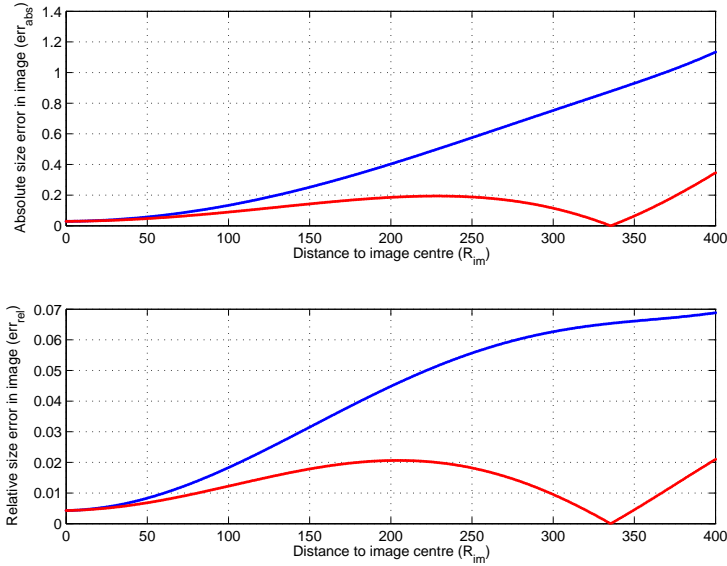


Figura 4.5: Error absoluto (arriba) y relativo (abajo) en función de la distancia al punto principal entre las longitudes de los semiejes obtenidas por simulación y las obtenidas por la formulación teórica. En azul, para el semieje mayor y en rojo para el semieje menor.

escala. Por lo tanto, en la implementación del nuevo parche para cámaras catadiópticas sólo se considerará el factor de escala completo para los puntos parametrizados en coordenadas cartesianas, ya que son aquellos en los que la incertidumbre de la profundidad tiene un valor suficientemente bajo.

Para los puntos que todavía tienen una alta incertidumbre en la profundidad puede contemplarse la aplicación del factor de escala, en función únicamente de la distancia del parche al punto principal  $R_{im}$ . Sin embargo, hay que tener en cuenta que en el caso de una cámara situada sobre un robot o la cabeza de una persona, ésta se mueve aproximadamente en un plano, por lo que el movimiento relativo de los puntos del entorno tiene lugar sobre planos paralelos. En la figura 4.6 se representa como evoluciona el tamaño de un objeto en la imagen en función de su distancia al punto principal cuando este se encuentra sobre planos a diferentes distancias  $z$  del plano en el que se mueve la cámara. Puede observarse que para objetos situados por debajo de la cámara ( $z < 0$ ), su tamaño decrece hasta llegar a 0 en el radio que define la línea de horizonte ( $R_\infty$ ); mientras que para objetos situados por encima ( $z > 0$ ) su tamaño crece a partir de  $R_\infty$ .

Como puede observarse en la figura 4.4 el término relacionado con la distancia al punto principal aumenta siempre con ésta, por lo que, en caso de no considerar la contribución de la profundidad, sólo tendría sentido su uso en parches que se proyectan en la imagen a una distancia  $R_{im} > R_\infty$ .

#### 4.3.3. Obtención del nuevo parche para emparejamiento

La computación del nuevo parche para emparejamiento se realizará mediante una transformación  $H_S$  de escala y rotación. Para construir dicha transformación se siguen



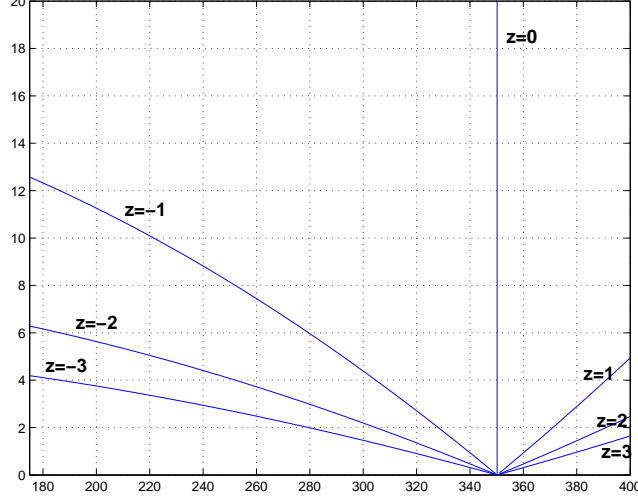


Figura 4.6: Evolución del tamaño de un parche en la imagen ( $r_{im}$ ) en función de su distancia al punto principal ( $R_{im}$ ) y de la distancia en metros ( $z$ ) del plano horizontal en el que se encuentra el parche en la escena respecto del plano en el que se mueve la cámara. Si el parche se encuentra por debajo de la cámara ( $z < 0$ ),  $r_{im}$  decrece con  $R_{im}$ , mientras que si se encuentra por encima  $r_{im}$  aumenta con  $R_{im}$ .

los siguientes pasos:

1) Comprobación de la condición  $k > \sqrt{2} \frac{h_P}{h_{BP}} \cos(\frac{\pi}{4} - \Delta\theta) + 0,1$ , donde  $h_P$  y  $h_{BP}$  son la mitad de la longitud del lado del parche descriptor y del parche grande redondeadas hacia abajo. Con esta condición se pretende evitar que el parche extraído no supere los límites del parche transformado. En el caso de que  $k$  no cumpla la condición se le asignará el valor límite.

2) Cálculo de la matriz de transformación:

$$H = H_{tr} H_S H_{tr}^{-1} \quad (4.17)$$

$$H_S = \begin{bmatrix} k \cos(\Delta\theta) & -k \sin(\Delta\theta) & 0 \\ k \sin(\Delta\theta) & k \cos(\Delta\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.18)$$

donde  $H_S$  es la matriz de transformación que combina la transformación de rotación  $R_{\Delta\theta}$  con el factor de escala  $k$  y  $H_{tr}$  es una matriz de traslación que traslada el sistema de coordenadas del parche de su esquina superior izquierda a su centro.

3) Obtención del parche transformado  $I_{WP}$  mediante el mapeo inverso de cada pixel al parche grande original  $I_{BP}$  y la realización de una interpolación bilineal.

$$\mathbf{X}_{BP} = H_S^{-1} \mathbf{X}_{WP} \quad (4.19)$$

$$I_{WP}(\mathbf{X}_{WP}) = \text{Interpolacion.bilineal}(I_{BP}, \mathbf{X}_{BP}) \quad (4.20)$$

4) Extracción del nuevo descriptor para emparejamiento del centro del parche transformado

#### 4.3. DESARROLLO DE UN NUEVO PARCHE INVARIANTE A ESCALA Y ROTACIÓN PARA IMÁGENES OMNIDIRECCIONALES

---

## Capítulo 5

# Realización de experimentos

En el presente proyecto nos hemos centrado por un lado en modificar el modelo de proyección para cámaras convencionales por el modelo de la esfera que es más general y válido para todo tipo de imágenes centrales y en particular para imágenes omnidireccionales catadióptricas. Por otro lado también nos hemos centrado en adaptar el parche descriptor a este tipo de cámaras para mejorar los resultados de emparejamiento.

En este capítulo se va a evaluar el resultado de aplicarle a los parches de los puntos característicos las transformaciones presentadas en el capítulo anterior. Para ello se han realizado 3 experimentos diferentes.

En primer lugar se han llevado a cabo dos experimentos con imágenes tomadas de la base de datos proporcionada por The Rawseeds Project <sup>1</sup> (Anexo B). De esta base de datos, se ha seleccionado una secuencia con más de 32000 imágenes adquiridas por un robot equipado con una cámara hipercatadióptrica. El primer experimento consiste en tres tests desacoplados del SLAM en los que se evalúa de forma separada la aplicación de la transformación de rotación y el factor de escala para el nuevo parche, comparándolo con un parche normal que permanece invariable desde su inicialización. En el segundo experimento se compara la diferencia de rendimiento en el SLAM al usar ambos tipos de parches a lo largo de una trayectoria.

El tercer experimento consiste en al ejecución del SLAM usando el nuevo parche con una secuencia tomada en el Campus Río Ebro de Zaragoza con una cámara omnidireccional acoplada a un casco desarrolladas dentro del proyecto VISPA <sup>2</sup>. Para evaluar los resultados obtenidos se tomará como referencia (*Ground Truth*) la trayectoria obtenida a partir de los datos proporcionados por un dispositivo GPS proporcionado por la Universidad de Zaragoza.

### 5.1. Experimento 1: Evaluación del proceso de emparejamiento utilizando el nuevo parche

En este primer experimento se han llevado a cabo tres tests donde se ha desacoplado el proceso de emparejamiento del algoritmo de la aplicación SLAM para hacer una evaluación preliminar de los atributos del nuevo parche comparándolo con un parche

---

<sup>1</sup>[HTTP://www.rawseeds.org](http://www.rawseeds.org)

<sup>2</sup>Non-conventional vision systems for personal assistance

## 5.1. EXPERIMENTO 1: EVALUACIÓN DEL PROCESO DE EMPAREJAMIENTO UTILIZANDO EL NUEVO PARCHES

---

al que no se le aplican transformaciones. En el primer y segundo test se ha evaluado la invarianza a rotación introducida por la nueva transformación bajo una rotación de la cámara de unos  $180^\circ$  y una traslación de la cámara con importantes cambios en las imágenes respectivamente. En el tercer test se ha evaluado el rendimiento de un parche al que se le aplica el factor de escala. El diseño de los tres tests es similar. En primer lugar, se han extraído los puntos característicos de la primera imagen de la secuencia seleccionada para el test mediante un extractor FAST. Posteriormente se han seleccionado algunos puntos de los extraídos y se han guardado sus correspondientes localizaciones y parches.

Como el proceso de emparejamiento ha sido desacoplado del SLAM, las localizaciones estimadas de los puntos característicos han sido seleccionadas manualmente en cada imagen y la región de búsqueda se ha fijado en un cuadrado de  $50 \times 50$  píxeles. El emparejamiento de cada punto se ha obtenido buscando el parche con mejor correlación dentro de la región de búsqueda, que es el mismo método que se usa en la aplicación SLAM.

Para cada punto característico e imagen se han definido las siguientes variables a evaluar:

- Correlación en la localización real del punto
- Mejor correlación en la región de búsqueda
- Distancia entre la localización real y la localización de la mejor correlación

### 5.1.1. Test 1: Transformación de rotación con rotación $180^\circ$ de la cámara

Siguiendo el procedimiento explicado anteriormente, para el primer test se ha seleccionado una secuencia en la que el robot rota  $180^\circ$ . De esta secuencia se han extraído 6 imágenes espaciadas entre sí por 20 imágenes y se han seleccionado 9 puntos característicos (figura 5.1).

En la figura 5.2 se muestran los resultados obtenidos para todos los parches. Los resultados de cada parche por separado así como una explicación más detallada de casos puntuales se muestran en el Anexo D. Como puede verse en la primera gráfica de la figura 5.1, la correlación en la localización real del punto es bastante mejor cuando se aplica la transformación de rotación que cuando no se aplica, lo que indica mayor robustez ante rotaciones de la cámara en torno al eje de revolución del espejo. La segunda y la tercera gráfica dan una idea del rendimiento de ambas opciones si estuviera integrado el proceso de emparejamiento en el algoritmo SLAM. La segunda gráfica muestra que es más probable el emparejamiento de los parches rotados, ya que obtienen mejores valores para la mejor correlación dentro de la región de búsqueda (todos por encima de 0,9). A partir de la tercera gráfica se concluye además, que la aplicación de una transformación de rotación disminuye la posibilidad de que los emparejamientos obtenidos sean falsos ya que la distancia entre los puntos con mejor correlación y la localización real del punto característico es muy pequeña, mientras que no usando la transformación la distancia es demasiado grande.

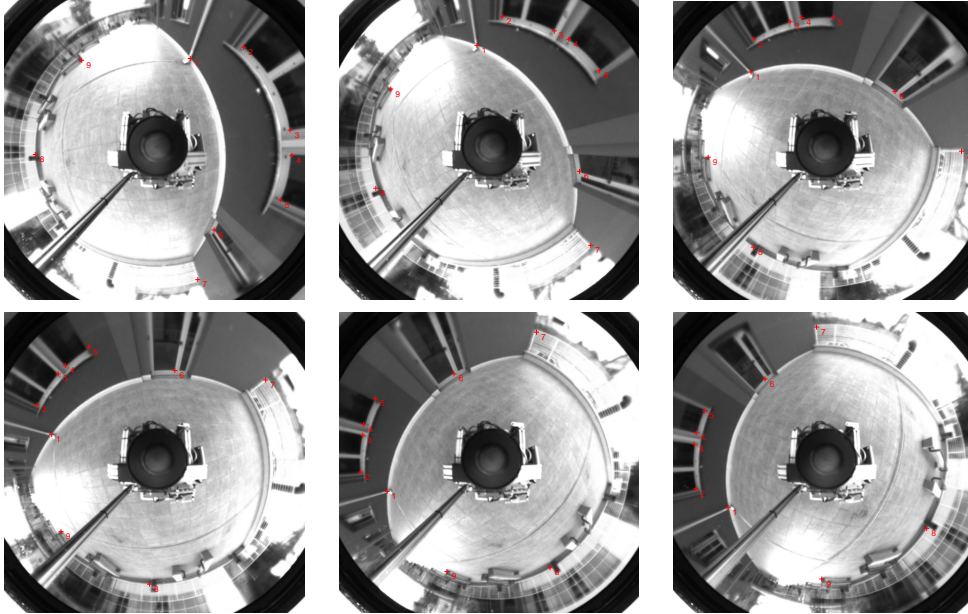


Figura 5.1: Secuencia de imágenes tomada para el test 1 (Rotación de la cámara de  $180^\circ$ ). Los puntos seleccionados para emparejamiento están marcados en rojo

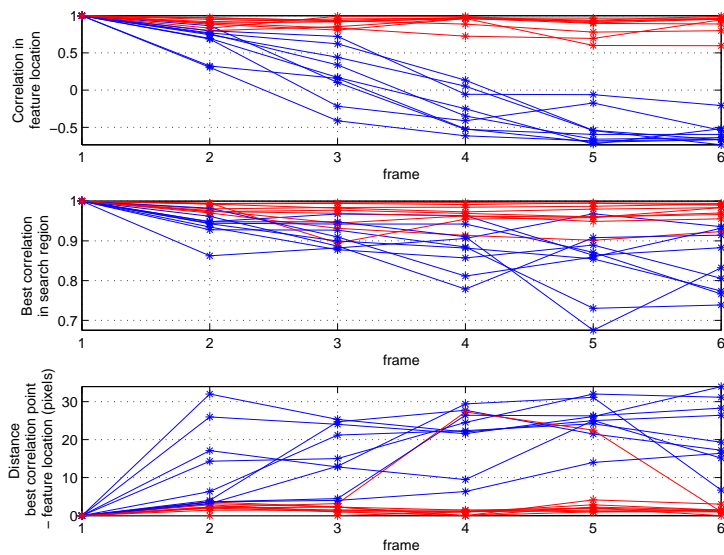


Figura 5.2: Resultados de emparejamiento para el test 1. En rojo, con transformación de rotación. En azul, sin transformación de rotación.

### 5.1.2. Test 2: Transformación de rotación con traslación de la cámara

Para el segundo test se ha seleccionado una secuencia que contiene sólo traslación de la cámara. De esta secuencia se han extraído 6 imágenes espaciadas entre sí por 20 imágenes y se han seleccionado 6 puntos característicos para hacer el emparejamiento (figura 5.1). Al igual que en el anterior test se evalúa la aplicación a los parches de una transformación de rotación antes del emparejamiento, pero esta vez ante un movimiento de traslación de la cámara.

En la figura 5.4 se muestran los resultados de todos los puntos característicos seleccionados a la vez (consultar Anexo D para ver las gráficas de cada punto por separado y una explicación más detallada). Éstos, al igual que en el test anterior muestran la superioridad de los parches a los que se les aplica la transformación de rotación, aunque en este caso la diferencia es menos apreciable que en el test anterior.

La razón es que al realizar la transformación de rotación se está suponiendo que el punto se mueve en la imagen sobre una circunferencia. Esta hipótesis es cierta en el caso de que exista únicamente rotación de la cámara en torno al eje de revolución del espejo, como en el test anterior. Sin embargo, en el caso de una traslación de la cámara, el movimiento relativo de los puntos del entorno 3D respecto de la cámara aparece en la imagen como un movimiento a lo largo de una curva cónica, cuya excentricidad aumenta desde 0 cuanto más alejado está el punto de la recta que define la dirección de movimiento de la cámara en la imagen. Cuando la excentricidad es 1, el punto se movería en la imagen sobre la circunferencia que marca al línea de horizonte, cuyo radio ( $R_\infty = \frac{z}{\xi}$ ) se obtiene de proyectar mediante el Modelo de la Esfera un punto cuya componente  $z$  en el sistema de referencia de la cámara tiende a 0.

De la intersección de la circunferencia con la recta a lo largo de la que se mueve la cámara resultan dos puntos de fuga de los que parten/llegan (dependiendo del sentido de traslación) las trayectorias que describen los puntos en la imagen durante la traslación.

Esto implica que, en un movimiento de traslación de la cámara, aplicar una transformación de rotación a los parches mejora el emparejamiento de puntos alejados de la trayectoria de la cámara, mientras que empeora el de puntos muy cercanos.

Los resultados muestran también que, a medida que la cámara se traslada, el emparejamiento tiende a hacerse de forma errónea tanto con parches rotados como con parches sin transformar. La razón es que en una traslación tienen lugar un cambio de escala y de punto de vista de los puntos característicos que no pueden ser predichos por la transformación de rotación.

### 5.1.3. Test 3: Factor de escala

En este test se lleva a cabo una evaluación del proceso de emparejamiento aplicando un cambio de escala a los parches mediante el factor de escala  $k$  formulado en la Sección 4.3.2. Dado que, como se explicó previamente, este experimento está desacoplado del SLAM, no es posible determinar la profundidad en la escena 3D de los parches extraídos, la cual es necesaria para calcular el factor de escala en un caso general. No obstante, interesa evaluar el cambio de escala introducido por la forma del espejo, ya que el introducido por el cambio de profundidad es simplemente inversamente proporcional a la misma. Por lo tanto en este test, no se considera la contribución de la profundidad y en lugar de seleccionar una secuencia de la base

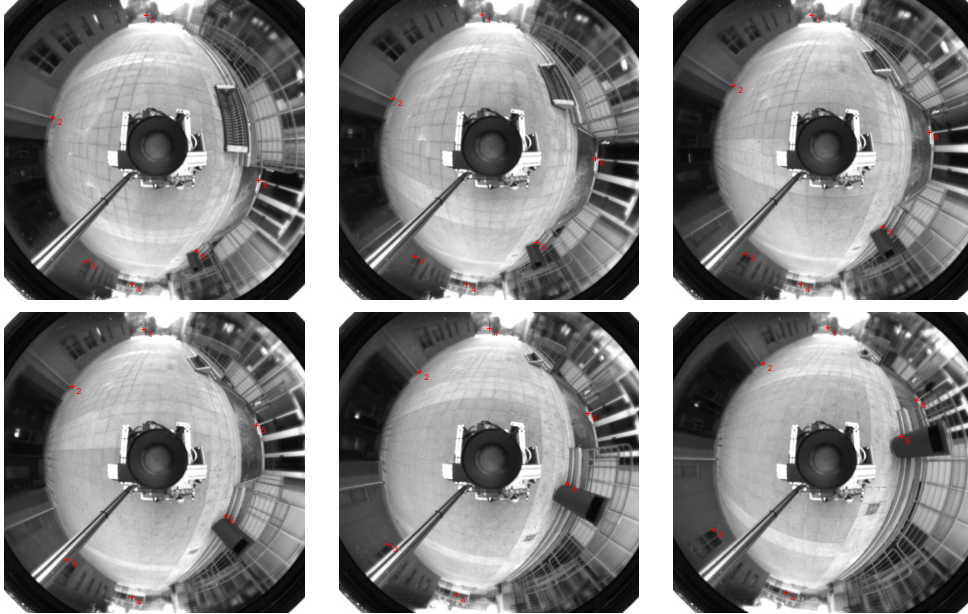


Figura 5.3: Secuencia de imágenes tomada para el test 2 (Traslación). Los puntos seleccionados para emparejamiento están marcados en rojo

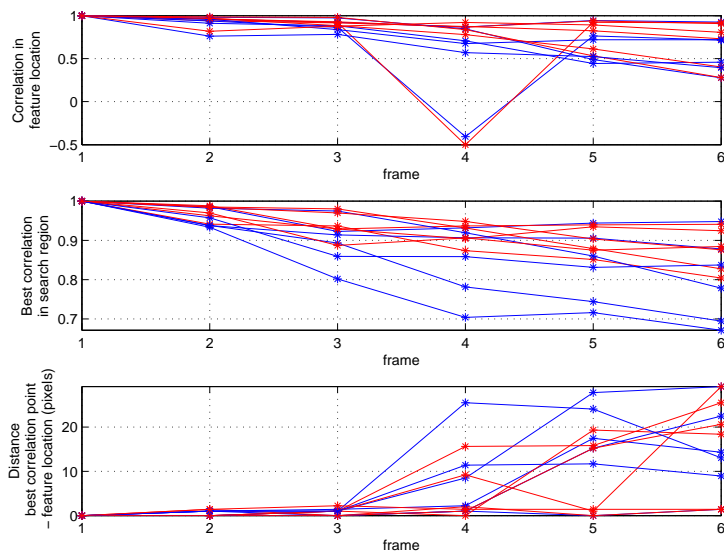


Figura 5.4: Resultados de emparejamiento para el test 2. En rojo, con transformación de rotación. En azul, sin transformación de rotación.

## 5.1. EXPERIMENTO 1: EVALUACIÓN DEL PROCESO DE EMPAREJAMIENTO UTILIZANDO EL NUEVO PARCHE

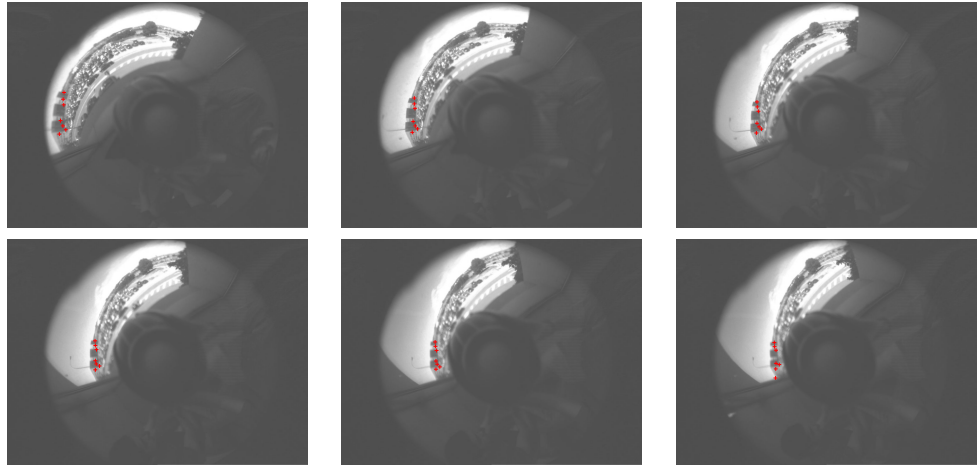


Figura 5.5: Secuencia de imágenes tomada para el 3 (Cambio de escala). Los puntos seleccionados para emparejamiento están marcados en rojo

de datos de Rawseeds, se ha hecho una adquisición de imágenes sin cambios de profundidad siguiendo los siguientes pasos:

- Seleccionar una zona en la escena con una potencial riqueza de puntos característicos y situado lo suficientemente lejos de la cámara de forma que se pueda asumir  $D \rightarrow \infty$ .
- Capturar imágenes mientras se rota la cámara de forma que la zona seleccionada se mueva solamente a lo largo de la dirección radial. Como se ha seleccionado una zona a una distancia infinita, no serán problemáticos posibles pequeños desplazamientos de la cámara durante la captura.

Siguiendo este procedimiento se tomó una secuencia de 6 imágenes para realizar el test, siendo 7 el número de puntos característicos seleccionados en una zona en la que no tiene lugar cambio de orientación en la imagen a lo largo de la secuencia (figura 5.5).

Dentro de este test se han considerado dos posibles casos: disminución de la escala ( $k < 1$ ) y aumento de la escala ( $k > 1$ ). En el caso de la disminución de la escala, la extracción de puntos característicos se realizó en la imagen donde la zona de extracción estaba lo más lejos posible del punto principal, mientras que para el caso de aumento de la escala el orden de la secuencia se invirtió y los puntos iniciales se extrajeron en la imagen en la que la zona de extracción estaba más cerca del punto principal.

En la figura 5.6 se muestran los resultados del caso de una disminución de la escala. Puede apreciarse que el hecho de aplicar un factor de escala frente a no hacerlo proporciona mejores valores de correlación y sobre todo una menor proporción de emparejamientos falsos al estimar de forma mucho más precisa la localización exacta del punto característico, como puede verse en la tercera gráfica.

Por el contrario, en el caso de un aumento de escala (figura 5.7 no existe apenas mejora al aplicar un factor de escala. La explicación más plausible es que cuando el parche de un punto característico aumenta su escala en la imagen aparece nueva información que no era visible cuando tenía una escala menor y que, por tanto, no es posible extraer del parche con menor resolución.



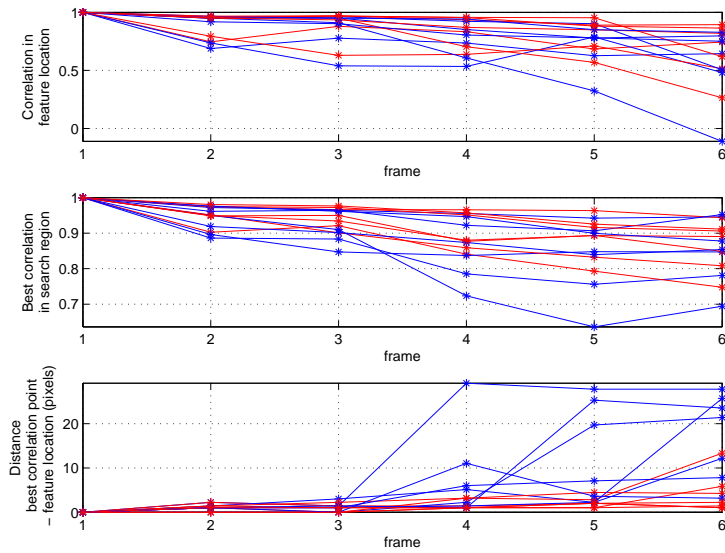


Figura 5.6: Resultados de emparejamiento para el test 3 (disminución de la escala). En rojo, con parche escalado. En azul, con parche sin escalar.

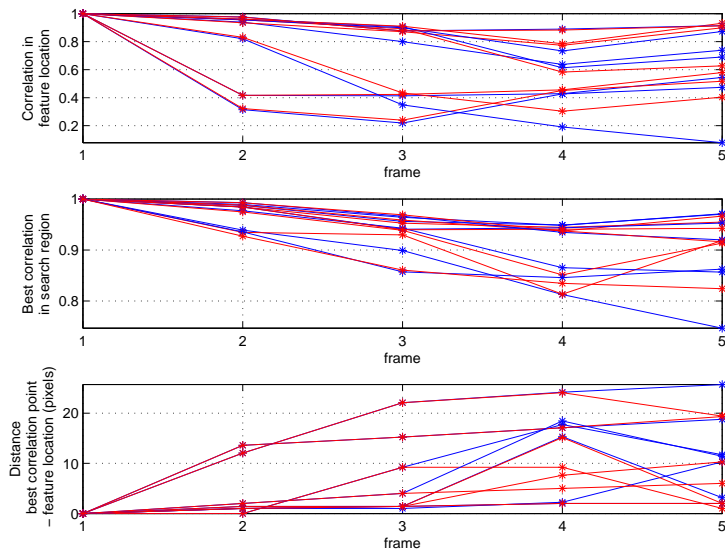


Figura 5.7: Resultados de emparejamiento para el test 3 (aumento de la escala). En rojo, con parche escalado. En azul, con parche sin escalar.

## 5.2. EXPERIMENTO 2: COMPROBACIÓN DE LA EJECUCIÓN DE UN SLAM CON EL NUEVO PARCHÉ

Tabla 5.1: Emparejamientos y puntos en el mapa final por punto inicializado

Umbral de correlación	Parche omnidireccional			Parche normal		
	$PI$	$R_e$	$R_m$	$PI$	$R_e$	$R_m$
0,8	8648	22,31	0,1	8923	19,75	0,087
0,9	9834	19,38	0,062	10854	16,09	0,046
0,95	13189	13,31	0,027	14970	10,56	0,019

### 5.2. Experimento 2: Comprobación de la ejecución de un SLAM con el nuevo parche

En el segundo experimento se ha evaluado la solución para SLAM con cámaras omnidireccionales presentada en este proyecto utilizando la aplicación en tiempo real. Para su evaluación se ha tratado de seleccionar una secuencia en exteriores de la base de datos proporcionada por el proyecto Rawseeds. Dicha selección ha resultado problemática ya que en algunas secuencias tomadas por la cámara omnidireccional, la imagen presentaba varias motas de suciedad lo que daba lugar a la detección de puntos inmóviles dentro de la zona no enmascarada provocando el colapso del SLAM. Por otro lado también ha sido necesario mejorar la calibración de la cámara proporcionada por Rawseeds (Anexo C). Finalmente se ha seleccionado una trayectoria en la que se cubre una distancia total de 360 metros.

Para comparar el nuevo parche con un parche normal al que no se le aplican transformaciones se ha ejecutado el SLAM en la secuencia seleccionada con diferentes umbrales de correlación para el emparejamiento y se han medido tres variables:

- Número total de puntos inicializados ( $PI$ )
- Ratio de emparejamientos por punto inicializado: ( $R_e = \frac{\text{Total emparejamientos}}{PI}$ )
- Ratio de puntos en el mapa por punto inicializado: ( $R_m = \frac{\text{Tamaño mapa final}}{PI}$ )

Los resultados de la tabla 5.1 muestran que el parche diseñado para cámaras omnidireccionales da lugar a un mejor rendimiento que el parche sin transformar, ya que se inicializan menos puntos característicos y se obtiene más información para el SLAM por punto inicializado. Esta diferencia es mayor a medida que se utiliza un umbral de correlación más exigente y por tanto, con menos errores de emparejamiento a priori. Por otra parte, la tabla también muestra que el umbral de correlación puede tener una influencia mayor en las variables mostradas que el tipo de parche utilizado, aunque a costa de disminuir la precisión a priori del SLAM.

En la figura 5.8 se muestran las proyecciones en los planos XY y YZ de la trayectoria obtenida con el nuevo parche y un umbral de correlación de 0,8. Nótese que aunque la aplicación MonoSLAM es capaz de estimar el movimiento de la cámara en el plano 3D, no estando limitada a un plano 2D, la trayectoria obtenida muestra que la cámara se está moviendo en un plano, como realmente ocurre, lo que da idea de su elevada precisión.

Finalmente hemos comparado esta trayectoria con el *Ground Truth* a partir de los datos del GPS, que también son proporcionados por Rawseeds. Debido al problema comentado anteriormente de la inobservabilidad de la escala en el SLAM, se ha escalado y alineado la trayectoria obtenida con la proporcionada por los datos del GPS (figura 5.9). Para evaluar numéricamente la precisión de la trayectoria obtenida

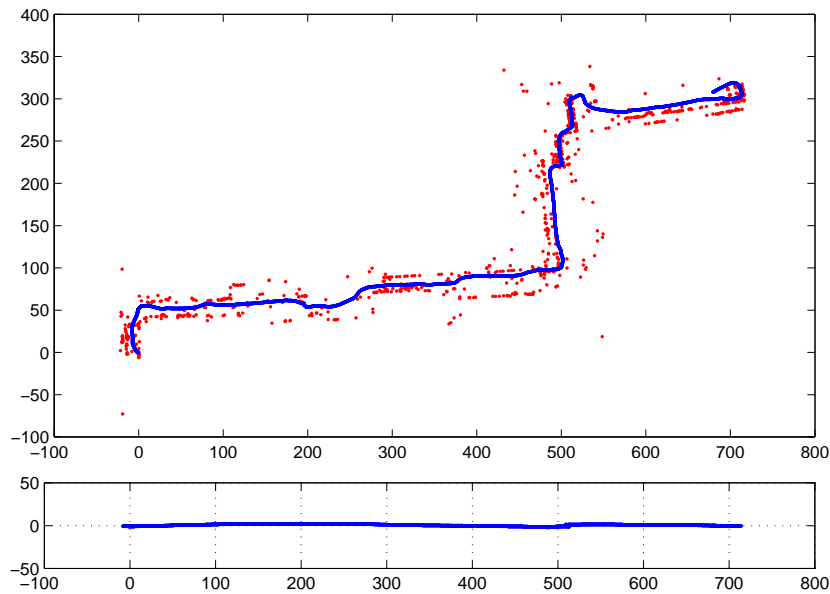


Figura 5.8: Trayectoria SLAM, obtenida con un umbral de correlación de 0,8 y usando el parche para cámara omnidireccional, proyectada en el plano XY (arriba) y en el plano XZ (abajo). Los puntos rojos corresponden a los puntos característicos en el mapa final. La figura sólo da una idea cualitativa de la trayectoria ya que las dimensiones dependen de una escala estimada de forma arbitraria, al no ser observable.

por la aplicación SLAM se han calculado el error medio de la distancia entre puntos correspondientes al mismo instante de tiempo de ambas trayectorias. El error medio calculado es  $\mu_{err} = 2,95 \text{ m}$  con una desviación típica de  $\sigma = 1,35 \text{ m}$  y un error máximo de  $max_{err} = 5,93 \text{ m}$ . Si dividimos el error medio entre la longitud de la trayectoria (360 m) obtenemos un error medio relativo del 0,85 %.

También se puede comprobar la exactitud del SLAM comparando la orientación estimada de la cámara con la calculada a partir de los datos del GPS por diferenciación (figura 5.10), con la ventaja de que en este caso no es necesario aplicar ningún factor de escala ya que es una variable que no depende de unidades de longitud.

### 5.3. Experimento 3: SLAM con casco-cámara

El tercer experimento consiste en la ejecución del SLAM usando el nuevo parche con secuencias tomadas en el Campus Río Ebro de Zaragoza con una cámara omnidireccional acoplada a un casco. Para evaluar los resultados obtenidos se tomará como referencia (*Ground Truth*) la trayectoria obtenida a partir de los datos proporcionados por un dispositivo GPS.

Durante la experimentación con el casco-cámara nos hemos encontrado con una serie de inconvenientes que no existían al trabajar con las secuencias de Rawseeds:

- *Baja repetitibilidad del intervalo de tiempo entre capturas:* La cámara permite 3 velocidades de adquisición (7,5, 15, y 30 fps). En principio se intentó realizar una adquisición a 15 fps pero el intervalo de tiempo entre capturas se desviaba

### 5.3. EXPERIMENTO 3: SLAM CON CASCO-CÁMARA



Figura 5.9: Trayectoria GPS (rojo) y trayectoria SLAM (verde) superpuestas sobre la imagen de satélite del Campus de Bovisa (Milán) donde fueron tomadas las secuencias por el equipo de Rawseeds

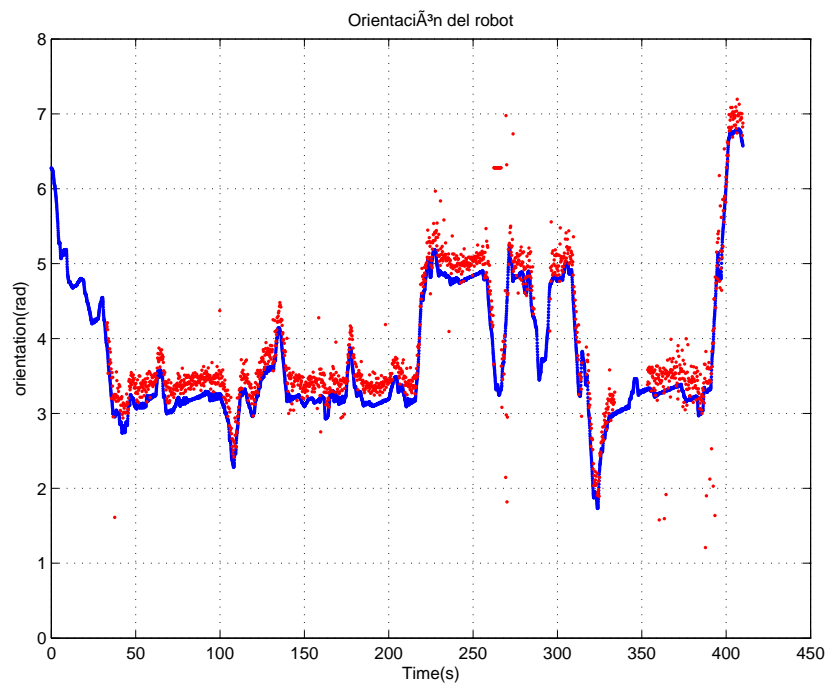


Figura 5.10: Orientación de la cámara obtenida del SLAM (azul) y de los datos de posición del GPS (rojo). El ruido en esta última se debe a que ha sido obtenida por diferenciación.

constantemente del valor nominal, por lo que se optó finalmente por una velocidad de adquisición de 7,5 fps con la que el intervalo de tiempo entre capturas durante la adquisición se ajusta a una distribución gaussiana centrada en el intervalo nominal y con una desviación típica relativamente baja.

- *Elevada vibración de la cámara:* Al ir sobre la cabeza de una persona, existen vibraciones debido al movimiento al andar o a posibles movimientos del cuello, lo que se traduce en un incremento del posible ruido de aceleración
- *Condiciones de iluminación:* Al hacerse las adquisiciones en el exterior durante días de verano se han encontrado dificultades a la hora de conseguir hacer una buena adquisición, debido a la gran variación de iluminación entre las zonas de luz y de sombra.

Para evaluar el rendimiento del SLAM con el casco cámara se han tomado dos secuencias: una corta, haciendo un recorrido en el espacio entre el edificio Ada Byron y la parada de autobús, y una larga con el objetivo de llevar el SLAM al límite, dando una vuelta completa alrededor los edificios Ada Byron, Torres Quevedo y Betancourt .

### 5.3.1. Secuencia corta

Para la secuencia corta se realizó la adquisición en un día soleado a lo largo de un recorrido de 236 m. Tras realizar algunas pruebas ajustando el umbral de correlación y otros parámetros de sintonía como ruidos de aceleración y ajustes del extractor de puntos, se han obtenido los resultados mostrados en las figuras 5.11 y 5.12.

La trayectoria obtenida mediante SLAM (figura 5.11) se ha escalado y alineado con la trayectoria obtenida a partir de los datos del GPS (figura 5.12). Debido a la imprecisión de las medidas del GPS en la zona cercana al estanque, solamente se ha considerado el tramo hasta ese punto para el escalado y posterior análisis cuantitativo de la precisión del SLAM. Asumiendo la referencia del GPS como *Ground Truth*, se ha calculado un error medio entre ambas trayectorias  $\mu_{err} = 3,03 m$  con una desviación típica de  $\sigma = 1,31 m$  y un error máximo de  $max_{err} = 5,82 m$ . Dividiendo el error medio entre la longitud del tramo considerado para el análisis (190 m) se ha obtenido un error medio relativo del 1,52 %.

AL igual que en el anterior experimento, en la figura 5.13 se compara también la orientación obtenida por el SLAM con la obtenida a partir de los datos del GPS.

### 5.3.2. Secuencia larga

La adquisición para la secuencia larga se realizó a lo largo de un recorrido de 1400 m durante un día nublado, para tener las mejores condiciones posibles de iluminación. Durante la ejecución de un SLAM con esta secuencia nos hemos encontrado con bastantes dificultades para conseguir que se complete todo el recorrido sin que colapse la aplicación. A los inconvenientes de la adquisición con el casco-cámara descritos anteriormente, hay que añadirle la dificultad para inicializar nuevos puntos en el mapa debido a la superficie con poca textura del suelo y de algunos edificios como el Betancourt, por lo que la cámara acaba perdiéndose en algún punto provocando el colapso de la aplicación.

No obstante se ha conseguido obtener una ejecución completa a base de reducir drásticamente el umbral de puntuación de la extracción FAST de puntos de la imagen. En la figura 5.14 se muestra la trayectoria SLAM estimada.

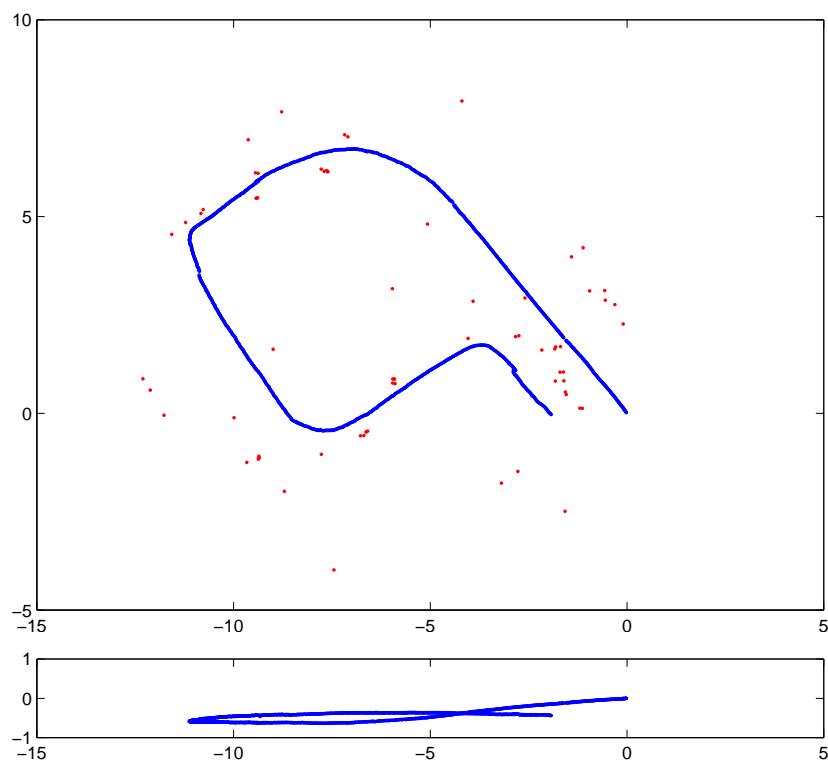


Figura 5.11: Trayectoria SLAM (secuencia corta con casco) proyectada en el plano XY (arriba) y en el plano XZ (abajo). Los puntos rojos corresponden a los puntos característicos en el mapa final. Las dimensiones dependen de una escala estimada de forma arbitraria, al no ser observable.



Figura 5.12: Trayectoria GPS (rojo) y trayectoria SLAM (verde) de la secuencia corta superpuestas sobre la imagen de satélite de la zona donde se hizo la adquisición. Dada la imprecisión del GPS a partir del punto marcado en negro en ambas trayectorias, se ha utilizado solamente el tramo hasta este punto para realizar el escalado y alineado de la trayectoria SLAM y el posterior análisis cuantitativo de la precisión.

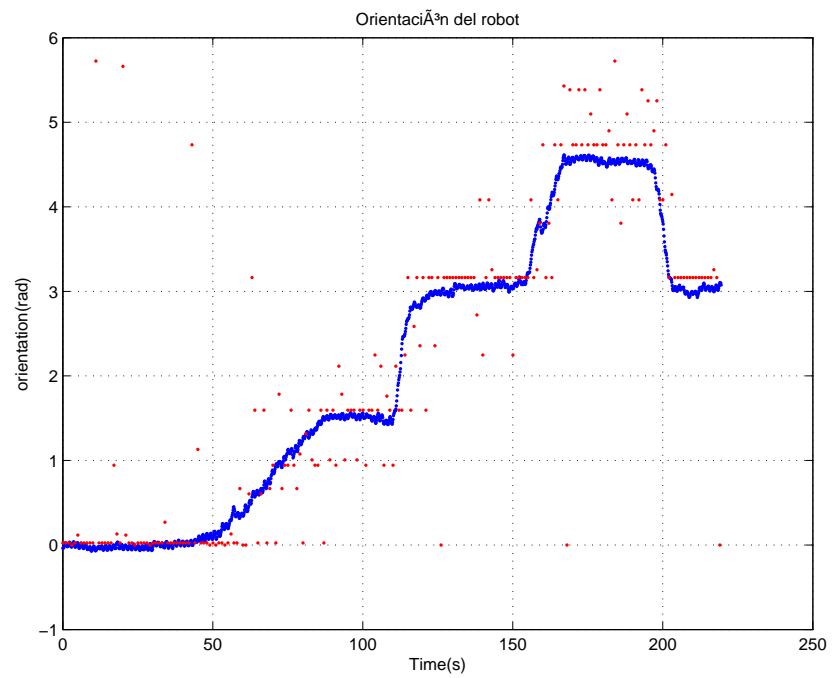


Figura 5.13: Orientación de la cámara obtenida del SLAM (azul) y de los datos de posición del GPS (rojo). Existe ruido en esta última ya que ha sido obtenida por diferenciación a partir de la posición dada por el GPS.



## CAPÍTULO 5. REALIZACIÓN DE EXPERIMENTOS

---

El alineado y escalado de la trayectoria con la obtenida de los datos del GPS ha sido problemático ya que, debido a la imprecisión del CPS por un lado, y a la gran longitud del recorrido por otro, el escalado de la trayectoria completa no proporcionaba buenos resultados. Por ello, al igual que en el caso anterior se ha escalado la trayectoria, considerando solo un tramo desde el punto de partida hasta un poco más allá de la mitad del recorrido, mostrándose el resultado en la figura 5.15.

Como era previsible en un recorrido tan largo, en este caso la trayectoria SLAM no se ajusta tan bien a la del GPS, aunque si que coinciden en forma. Aparte de la deriva natural del SLAM, existe una deriva de escala a lo largo del recorrido, por lo que un único escalado no es suficiente para ajustar la trayectoria. Una consecuencia evidente de la deriva de escala sería que en la trayectoria SLAM no se retorna al punto de partida.

En la figura 5.16 se muestra la estimación de la orientación de la cámara, la cual no depende de la escala, a lo largo del recorrido comparada con la calculada a partir de los datos del GPS. Obviando el ruido en la orientación estimada a partir de los datos del GPS, se observa que aunque en la aplicación SLAM se estima el cambio de orientación en las curvas de forma correcta, a lo largo de los dos grandes tramos en línea recta se va acumulando un error de orientación, lo que no ocurría en los anteriores experimentos realizados. Este efecto puede observarse también de forma clara en las figuras 5.14 y 5.15 donde en los tramos rectos, la cámara se desvía constantemente hacia su derecha. La explicación a este fenómeno puede ser la presencia de diferentes escalas alrededor de la cámara en una misma posición, lo cual afecta considerablemente a las cámaras omnidireccionales por su gran ángulo de visión: si los puntos a un lado de la cámara se mueven poco en la imagen por estar muy alejados de la cámara y los del otro lado se mueven más por estar más cerca, la aplicación percibe de forma errónea, debido a la inobservabilidad de la profundidad en un SLAM monocular, que se está describiendo una rotación en torno a los puntos que permanecen prácticamente inmóviles.

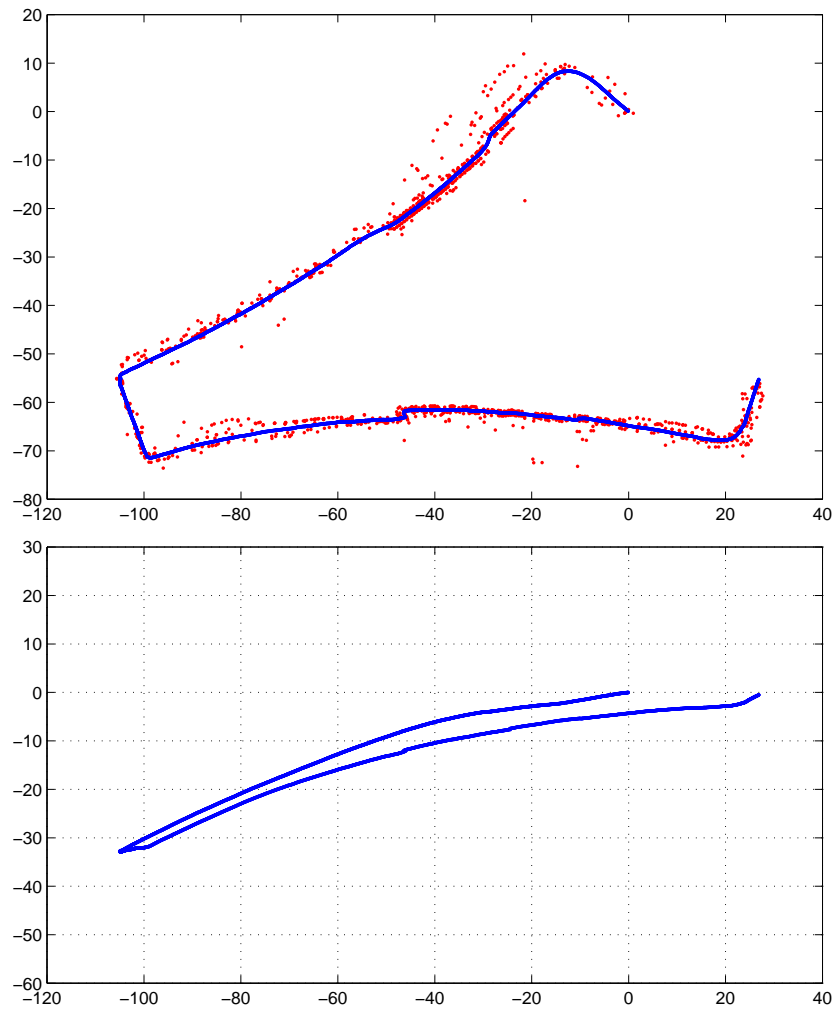


Figura 5.14: Trayectoria SLAM (secuencia larga con casco) proyectada en el plano XY (arriba) y en el plano XZ (abajo). Los puntos rojos corresponden a los puntos característicos en el mapa final

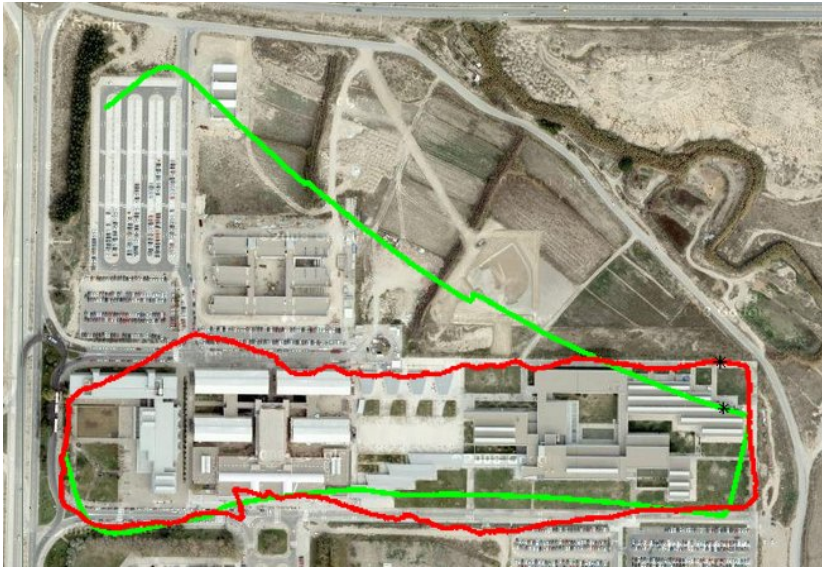


Figura 5.15: Trayectoria GPS (rojo) y trayectoria SLAM (verde) de la secuencia larga superpuestas sobre la imagen de satélite de la zona donde se hizo la adquisición. El escalado de la trayectoria se ha realizado con el tramo desde el inicio hasta el punto marcado en negro en ambas trayectorias.

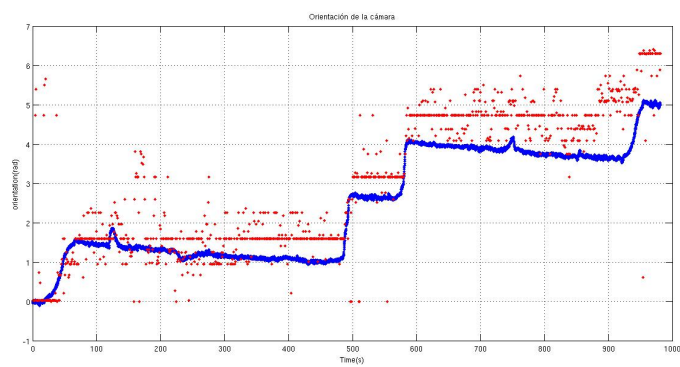


Figura 5.16: Orientación GPS (rojo) y orientación SLAM (verde) de la cámara durante la secuencia larga.

### 5.3. EXPERIMENTO 3: SLAM CON CASCO-CÁMARA

---

## Capítulo 6

# Conclusiones

El objetivo de este proyecto de fin de carrera ha sido desarrollar un sistema de SLAM monocular en tiempo real con visión omnidireccional. Para ello, se ha partido de una aplicación SLAM basada en EKF ya existente para cámaras convencionales implementada en C++ por Andrew J. Davison <sup>1</sup>. Se trata de una aplicación de software libre y código abierto que ha sido continuamente actualizada desde su lanzamiento con la colaboración de varios autores entre los que se encuentran J.M.M. Montiel y Javier Civera de la Universidad de Zaragoza, que han introducido mejoras importantes como la parametrización por profundidad inversa ([9]) para los puntos recién inicializados o la implementación de un filtro RANSAC ([8]) de un punto para realizar el filtrado de mediciones.

La principal motivación para afrontar el proyecto ha sido la mayor ventaja que, gracias a su amplio ángulo de visión, ofrecen las cámaras omnidireccionales frente a las cámaras proyectivas al tratar de resolver un problema SLAM, lo cual se había demostrado previamente en el trabajo desarrollado por Alejandro Rituerto ([23])

Para alcanzar el objetivo propuesto se han realizado una serie de modificaciones en el código original. Las principales modificaciones realizadas han sido de dos tipos.

En primer lugar, el modelo de proyección se ha adaptado a uno apropiado para sistemas omnidireccionales. Se ha implementado el llamado Modelo de la Esfera, un modelo no lineal que permite calcular la proyección de puntos del espacio en puntos de la imagen omnidireccional, y de puntos en la imagen en rayos tridimensionales en los que se encuentra el punto. Se trata de una modificación necesaria para el correcto funcionamiento del SLAM con cámaras omnidireccionales.

Por otro lado, el descriptor de puntos característicos permite identificar los puntos en una imagen, necesario para poder hacer mediciones mediante emparejamientos de los puntos en una imagen nueva con los puntos del mapa. Las imágenes omnidireccionales conllevan un modelo de proyección más complejo, así como una importante deformación y una escala variable en la imagen debido al cambio de resolución a lo largo de la dirección radial. Por lo tanto, la segunda modificación es la implementación de un nuevo descriptor específico para cámaras omnidireccionales que sea invariante a estos cambios de escala e invariante a rotación de la cámara respecto al eje de revolución del espejo. Esta modificación es opcional, ya que el algoritmo SLAM permite su ejecución de manera razonable con un descriptor normal sin invarianza a rotación ni cambios de escala. Por ello, en la fase de experimentación se ha llevado a

---

<sup>1</sup><http://www.doc.ic.ac.uk/~ajd/software.html>

---

cabo una comparativa entre ambos tipos de descriptores para evaluar el grado de mejora introducido por el nuevo descriptor implementado.

También se han realizado otros pequeños cambios como la modificación del proceso de filtrado de puntos medidos, adaptándolo al nuevo modelo de proyección y a la morfología de las imágenes omnidireccionales, y la marginalización fuera del vector de estados de puntos del mapa antiguos que dejan de ser detectados, con el objetivo de reducir el coste computacional cuando el mapa alcanza un tamaño considerable. Además se ha corregido un fallo de la aplicación en el proceso de borrado de puntos no fiables, ya que no se eliminaban todos los que debían eliminarse en cada paso del SLAM.

Una vez realizados los cambios en la aplicación original se ha procedido a la experimentación. Los experimentos se han realizado en dos fases.

La primera fase consta de dos experimentos en los que se han utilizado datos facilitados por el proyecto Rawseeds para comparar el nuevo descriptor específico para cámaras omnidireccionales y un descriptor normal. En el experimento 1, se ha realizado una comparación del proceso de emparejamiento desacoplado del SLAM bajo diferentes condiciones (rotación, traslación, cambio de escala), comprobándose que el nuevo descriptor mejora en prestaciones al descriptor normal.

En el experimento 2, se ha comparado el rendimiento de ambos descriptores durante la ejecución de un SLAM a lo largo de una trayectoria de 360 metros. Los resultados muestran que el nuevo parche ofrece un mejor rendimiento, al ser necesario inicializar menos puntos característicos y que se consigue un mayor número de emparejamientos exitosos por punto inicializado. Se ha comparado además una de las trayectorias obtenidas con el nuevo parche, con la trayectoria dada por el GPS considerada como *Ground Truth*. Debido a la inobservabilidad de la escala y a las distintas referencias de orientación de SLAM y GPS se ha realizado un alineamiento y escalado de la trayectoria SLAM para ajustarla a la del GPS. También se han obtenido las estimaciones de orientación de la cámara del SLAM y del GPS. Los resultados finales muestran la elevada precisión del SLAM.

En la segunda fase se ha experimentado con dos secuencias, una corta y una larga tomadas por la cámara-casco en el Campus Río Ebro, utilizando la nueva implementación para el parche. Se han encontrado dificultades para realizar buenas adquisiciones debido a diferentes factores como tiempos de adquisición erráticos, vibraciones de la cámara o una iluminación no uniforme de la escena.

La secuencia corta se tomó en el espacio entre el edificio Ada Byron y la parada de autobús a lo largo de un recorrido de 236 metros. Tras ejecutar el SLAM, se han comparado las estimaciones de trayectoria y orientación con las obtenidas a partir de los datos del GPS, procediendo del mismo modo que en el anterior experimento. Dada la imprecisión del GPS hacia el final del recorrido sólo ha sido posible realizar la comparación en un tramo de 190 metros. Los resultados muestran una gran precisión del SLAM en la estimación de la trayectoria y de la orientación.

La toma de la secuencia larga se hizo a lo largo de un recorrido de 1400 metros rodeando los edificios Ada Byron, Torres Quevedo y Betancourt. Tras realizar el escalado y alineado, la estimación de la trayectoria del SLAM muestra una gran desviación respecto a la del GPS. La posible causa de esta desviación se estableció, además de en la deriva del error propia de un algoritmo incremental como el EKF-SLAM, en la deriva de escala que puede existir a lo largo del recorrido por su longitud y que hace insuficiente la aplicación de un escalado para el ajuste de la trayectoria. Sin embargo, evaluando la estimación de la orientación, que no depende de la escala y por lo tanto no debería verse afectada por la deriva de escala a lo largo del recorrido, se

## CAPÍTULO 6. CONCLUSIONES

---

ha observado que existe una pequeña variación de orientación a lo largo de los tramos rectos, que hace que se acumule un error de orientación. Este fenómeno nos lleva a pensar que, además de la ya mencionada deriva de escala a lo largo del recorrido existe otra fuente de imprecisión en la presencia de diferentes escalas alrededor de una misma localización, que sería un problema propio de las cámaras omnidireccionales por su gran ángulo de visión.

---



## Anexo A

# Parametrización de profundidad inversa

Uno de los mayores problemas del SLAM monocular es la inicialización de nuevos puntos característicos en el mapa, concretamente en la estimación inicial de la incertidumbre de la profundidad para puntos distantes. En una parametrización cartesiana según los ejes XYZ, las incertidumbres iniciales en la posición para puntos con poco paralaje no se ajustan a una distribución Gaussiana implícita en el EKF, por lo que antes de su inicialización es necesario extraer información de varias imágenes para poder obtener una estimación precisa de la profundidad. Esta inicialización especial conlleva el uso de algoritmos que se desvían del uso de distribuciones de probabilidad Gaussianas.

La parametrización de profundidad inversa, desarrollada por J.Civera y J.M.Montiel en [9] permite cubrir la incertidumbre de profundidad en un rango de profundidades que comprende desde un punto cercano a la cámara hasta el infinito, expresando el inverso de la profundidad como una distribución Gaussiana en lugar de la propia profundidad. Esta parametrización da lugar por tanto a una representación unificada en la que no es necesario un tratamiento especial a los puntos recién seleccionados para inicialización, de forma que se pueden inicializar y poner a disposición del algoritmo SLAM en el mismo momento de su selección.

Con la nueva parametrización, un punto 3D  $i$  se puede representar mediante un vector de 6 estados:

$$\mathbf{y}_i = (x_i, y_i, z_i, \theta_i, \phi_i, \rho_i) \quad (\text{A.1})$$

donde  $(x_i, y_i, z_i)$  son las las coordenadas de la cámara en el momento de la inicialización del punto,  $\theta_i$  y  $\phi_i$  son los ángulos de elevación y azimuth respectivamente y  $\rho_i$  es el inverso de la profundidad.

El paso de parametrización IDP a parametrización cartesiana se realiza mediante la siguiente ecuación:

$$\mathbf{y}_{XYZ} = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \frac{1}{\rho_i} \mathbf{m}(\theta_i, \phi_i) \quad (\text{A.2})$$

donde  $\mathbf{m} = (\cos \phi_i \sin \theta_i, -\sin \phi_i, \cos \phi_i \cos \theta_i)$

Como el modelo de proyección presentado en el Capítulo 2 se realiza la proyección a partir de un punto en coordenadas homogéneas también puede ser conveniente obtener el cambio de IDP a coordenadas homogéneas:

$$\mathbf{y}_h = \begin{pmatrix} X_{h,i} \\ Y_{h,i} \\ Z_{h,i} \\ W_{h,i} \end{pmatrix} = \begin{pmatrix} \rho_i \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \mathbf{m}(\theta_i, \phi_i) \\ \rho_i \end{pmatrix} \quad (\text{A.3})$$

### A.1. Inicialización de puntos característicos

El estado inicial para un punto recién observado viene dado por:

$$\mathbf{y}_i(\mathbf{r}^{WC}, \mathbf{q}^{WC}, \mathbf{p}, \rho_0, \sigma_{\rho_0}) = (x_i, y_i, z_i, \theta_i, \phi_i, \rho_i) \quad (\text{A.4})$$

donde  $\mathbf{r}^{WC}$  y  $\mathbf{q}^{WC}$  indican la posición y orientación de la cámara,  $\mathbf{p}$  la observación del punto en la imagen y  $\rho_0$  y  $\sigma_{\rho_0}$  las estimaciones iniciales de la media y la desviación típica para la distribución gaussiana de la profundidad inversa.

La posición de la cámara en el momento de la inicialización se obtiene de forma trivial:

$$(x_i, y_i, z_i) = \mathbf{r}^{WC} \quad (\text{A.5})$$

Los ángulos de azimuth y elevación se obtienen del rayo proyectivo  $\mathbf{x}$  en el que es mapeado el punto  $\mathbf{p}$  según el modelo inverso de proyección, expresado en la referencia absoluta:

$$\mathbf{m}(\theta_i, \phi_i) = \mathbf{R}^{CW} \frac{\mathbf{x}(\mathbf{p})}{\|\mathbf{x}(\mathbf{p})\|} \quad (\text{A.6})$$

$$\begin{pmatrix} \theta_i \\ \phi_i \end{pmatrix} = \begin{pmatrix} \arctan(m_x, m_z) \\ \arctan(-m_y, \sqrt{m_x^2 + m_z^2}) \end{pmatrix} \quad (\text{A.7})$$

A la profundidad inversa  $\rho_i$  se le asigna un valor inicial  $\rho_0$  determinado empíricamente junto con su desviación típica  $\sigma_{\rho_0}$  de forma que el intervalo de confianza del 95 % englobe un rango de profundidades desde la cercanía de la cámara hasta el infinito. Por ello la desviación típica debe ser elegida de tal forma que uno de los límites del intervalo de confianza sea negativo.

La covarianza del resto de variables de estado del punto se obtienen de la covarianza  $\mathbf{R}_i$  del error de medida y de la covarianza del estado total  $\mathbf{P}_{k|k}$ , siendo la covarianza del estado después de la inicialización:

$$\mathbf{P}_{k|k}^{new} = \mathbf{J} \begin{bmatrix} \mathbf{P}_{k|k} & 0 & 0 \\ 0 & \mathbf{R}_i & 0 \\ 0 & 0 & \sigma_{\rho_0} \end{bmatrix} \mathbf{J}^T \quad (\text{A.8})$$

$$\mathbf{J} = \left[ \begin{array}{c|c} \mathbf{I} & 0 \\ \hline \frac{\partial \mathbf{y}}{\partial \mathbf{r}^{WC}}, \frac{\partial \mathbf{y}}{\partial \mathbf{q}^{WC}}, 0, \dots, 0 & \frac{\partial \mathbf{y}}{\partial \mathbf{p}}, \frac{\partial \mathbf{y}}{\partial \rho} \end{array} \right] \quad (\text{A.9})$$

## A.2. Cambio de IDP a parametrización XYZ

El inconveniente de la parametrización de profundidad inversa es el aumento de coste computacional que supone expresar el estado de un punto con 6 parámetros en vez de los 3 de la parametrización cartesiana. Para hacer frente a este problema, se trata de convertir los puntos, siempre que sea posible, de IDP a parametrización XYZ. La posibilidad de realizar esta transformación vendrá dada por el valor de un índice de linealidad  $L_d$ , que tiende a 0 cuanto más se aproxima la profundidad del punto a una distribución Gaussiana. Dicho índice se determina de la siguiente manera:

$$L_d = \frac{4\sigma_d}{d_i} |\cos \alpha| \quad (\text{A.10})$$

con  $\sigma_d = \frac{\sigma_\rho}{\rho_i}$ ,  $d_i = \|\mathbf{h}_{XYZ}^W\|$ ,  $\alpha = \mathbf{m}^\top \mathbf{h}_{XYZ}^W \|\mathbf{h}_{XYZ}^W\|^{-1}$   
 $\sigma_\rho$  es la covarianza extraída de la matriz de covarianzas del estado del punto que se quiere convertir y  $\mathbf{h}_{XYZ}^W = \mathbf{y}_{XYZ} - \mathbf{r}^{WC}$  donde  $\mathbf{y}_{XYZ}$  son las coordenadas del punto en parametrización cartesiana obtenidas mediante la ecuación A.2.

Si  $L_d$  se encuentra por debajo del umbral de conversión determinado experimentalmente, la transformación se realiza mediante la ecuación A.2 para el estado del punto y mediante el correspondiente jacobiano para la covarianza:

$$\mathbf{P}^{new} = \mathbf{J}\mathbf{P}\mathbf{J}^\top \quad (\text{A.11})$$

$$\mathbf{J} = \text{diag}\left(\mathbf{I}, \frac{\partial \mathbf{y}_{XYZ}}{\partial \mathbf{y}_{IDP}}, \mathbf{I}\right) \quad (\text{A.12})$$

## A.2. CAMBIO DE IDP A PARAMETRIZACIÓN XYZ

---

## Anexo B

# La base de datos de Rawseeds

Las imágenes utilizadas para la realización de algunos de los experimentos del presente proyecto han sido tomadas de la base de datos puesta a disposición por el proyecto Rawseeds, cuyo objetivo es construir un conjunto exhaustivo de herramientas de referencia para sistemas robóticos compuesto por:

- Conjunto de datos de alta calidad tomados con diferentes tipos de sensores y con un *Ground Truth* asociado
- Problemas de referencia basados en los conjuntos de datos disponibles
- Soluciones de referencia para los problemas

Este conjunto de herramientas está dirigido sobre todo a la resolución de problemas de localización, mapeado y SLAM en robótica, aunque su uso no está estrictamente limitado a ello, y están disponibles para descargar libremente en la página web de Rawseeds.

El conjunto de datos del proyecto Rawseeds corresponde a los datos tomados de forma sincronizada por el conjunto de sensores equipados en un robot mientras éste explora un entorno. Cada conjunto de datos está compuesto por un conjunto de sesiones, cada una de las cuales incluye los datos tomados en la misma zona en momentos diferentes. Todos los conjuntos de datos de Rawseeds contienen información de varios tipos de sensores, incluyendo datos precisos sobre la trayectoria de referencia (*Ground Truth*). El *Ground Truth* es necesario para evaluar el rendimiento de cualquier solución de referencia aplicada a los datos de uno o varios sensores.

El robot utilizado para la adquisición de los datos ha sido construido por el propio equipo de Rawseeds, quienes lo han bautizado como Robocom (figura B.1). Este robot se mueve mediante tracción diferencial, y dispone de los siguientes sensores:

- Sensores ultrasónicos de profundidad alrededor del robot.
- Sensores láser de profundidad.
- Un sensor inercial que facilita la orientación del robot en 3 ejes.
- Sistemas de visión de diferentes tipos. Dispone de sistemas binocular y trinocular, de una cámara perspectiva convencional y un sistema hiper-catadióptrico de visión omnidireccional.



Figura B.1: Robot Robocom con el que se han adquirido los datos

Además de todos estos sistemas de medición, la trayectoria de referencia o *Ground Truth* es recogida mediante odometría y GPS.

Para este proyecto se han utilizado las imágenes tomadas por la cámara omnidireccional de una sola sesión correspondiente a una secuencia de datos tomada en el campus politécnico de Milán, en Italia. La secuencia completa cuenta con 32222 imágenes omnidireccionales, capturadas a 15 imágenes por segundo. La trayectoria seguida durante dicha secuencia es mixta, atravesando tanto tramos al aire libre como tramos por el interior de los edificios.

Para determinar la trayectoria de referencia se ha optado por utilizar la información aportada por el GPS, ya que, con un error máximo de posicionamiento de  $2cm$ , es mucho más preciso que la odometría del robot. Como la señal del GPS no se puede recibir en los tramos interiores, para la realización del experimento con SLAM se ha seleccionado el tramo exterior más largo dentro de la sesión de adquisición elegida, que consta de 6151 imágenes omnidireccionales tomadas a lo largo de una trayectoria de  $340m$ .

## Anexo C

# Calibración de las cámaras

En este anexo se presentan los datos de calibración de las cámaras con las que se han extraído las imágenes para este proyecto: la cámara omnidireccional del robot de Rawseeds y la cámara-casco del Grupo de Robótica, Percepción y Tiempo Real de la Universidad de Zaragoza. La calibración de ambas cámaras se ha llevado a cabo mediante la aplicación en Matlab desarrollada por Mei [20], mostrándose en la tabla C.1 los parámetros obtenidos para cada cámara.  $x_c$  y  $y_c$  son las coordenadas del centro de la imagen,  $\gamma_x$  y  $\gamma_y$  son las distancias focales generalizadas en cada dirección de la imagen,  $\xi$  es el parámetro característico del Modelo de la Esfera y  $k_c$  los parámetros que modelan la distorsión en la imagen.

	Cámara Rawseeds	Casco-cámara
Resolución	640x640	1024x768
$x_c$	325,56	516,8172
$y_c$	313,88	414,0827
$\gamma_x$	266,23	326,9527
$\gamma_y$	266,94	325,1058
$\xi$	0,93	0,9337
$k_c$	$1,9283 \times 10^{-3}$ $-5,0132 \times 10^{-3}$ $-4,5787 \times 10^{-4}$ $-3,8018 \times 10^{-4}$ 0	$-9,2582 \times 10^{-2}$ $1,7376 \times 10^{-2}$ $2,4574 \times 10^{-4}$ $-2,1840 \times 10^{-3}$ 0

Tabla C.1: Datos de calibración de las cámaras omnidireccionales

---



## Anexo D

# Resultados completos de los tests del experimento 1

En el Capítulo 5 se describe de forma precisa el diseño del primer experimento realizado. Para dicho experimento se han utilizado secuencias de imágenes tomadas de la base de datos proporcionada gratuitamente por Rawseeds. También se ha utilizado una secuencia tomada con un casco cámara. El experimento consiste en tres tests desacoplados del SLAM en los que se evalúa de forma separada la aplicación de la transformación de rotación y el factor de escala para el nuevo parche antes del proceso de emparejamiento, comparándolo con el emparejamiento realizado con un parche normal que permanece invariable desde su inicialización.

Para ello se han definido las siguientes variables para medir por cada imagen y punto característico seleccionado para el experimento.

- Correlación en la localización real del punto
- Mejor correlación en la región de búsqueda
- Distancia entre la localización real y la localización de la mejor correlación

En el primer test, se ha seleccionado una secuencia en la que el robot rota  $180^\circ$  compuesta de 6 imágenes, tomando 9 puntos del entorno para el análisis. Los resultados obtenidos se muestran en la figura D.1 para cada punto por separado.

Se van a analizar los resultados obtenidos para los parches 4 y 5, el primero por ser bastante representativo de lo que ocurre también con el resto de parches, y el segundo por poner de manifiesto un comportamiento especial. Como puede observarse en las gráficas del punto 4, el emparejamiento utilizando parches invariantes a rotación mejora notablemente respecto al obtenido con parches definidos en coordenadas cartesianas, siendo la correlación tanto en la localización estimada como en la de mayor correlación prácticamente 1 y con un error mínimo de localización como puede verse en la tercera gráfica. Nótese además, que en el caso de utilizar un parche sin rotar, la correlación en la localización estimada en la última imagen es negativa, consecuencia de que se obtiene la correlación entre parches prácticamente iguales pero con una diferencia de orientación de  $180^\circ$ .

Los resultados para el punto 5 sirven para resaltar lo que ocurre cuando un punto queda oculto en algún momento durante la secuencia. Como puede observarse en la secuencia de imágenes, el punto 5 corresponde a una esquina de una ventana que

---

durante los fotogramas 4 y 5 queda oculta tras la pared. En las gráficas puede observarse que la correlación en la localización estimada está alrededor de 0,7, la cual es bastante pobre respecto a las que se obtienen con los demás puntos. Sin embargo, se observa que en la región de búsqueda se obtiene una mejor correlación de 0,95, pero corresponde a un emparejamiento erróneo ya que como puede verse esta separada más de 20 píxeles de la localización estimada.

Finalmente para poder comparar de un vistazo general ambos tipos de parches se representan en la misma gráfica todos los datos obtenidos (figura D.2).

Para el segundo test, la secuencia seleccionada comprende un tramo con movimiento de traslación de la cámara con 6 imágenes, habiéndose seleccionado 6 puntos para el test. Los resultados para cada punto se muestra en la figura D.3. En la figura D.4 se muestran los resultados para todos los puntos de forma conjunta.

Los resultados para los puntos 1 y 4 son bastante similares y muestran que apenas existen diferencias entre el parche invariante a rotación y el parche normal. La razón es que, como se puede ver en la secuencia de imágenes, ambos puntos apenas cambian de posición durante la misma ya que se encuentran bastante cerca de los puntos de fuga dados por la circunferencia que marca la línea de infinito y la recta de la dirección de movimiento de la cámara.

En los resultados para el punto 6 se percibe una superioridad clara del parche con transformación de rotación, ya que, como puede verse en las imágenes de la cámara, su trayectoria en la imagen se aproxima bastante a la de una circunferencia. Esto se percibe también, aunque en menor medida, en los puntos 2, 3 y 5, donde la menor excentricidad de sus trayectorias hace que la superioridad del parche invariante a rotación no sea tan clara. Además también puede comprobarse en el caso de estos últimos puntos el efecto del cambio del punto de vista durante la traslación, que da lugar a errores de medida en las últimas imágenes de la secuencia usando ambos tipos de parche.

Para el tercer test se ha realizado una adquisición con el casco-cámara en las condiciones detalladas en el Capítulo 5, con el fin de evaluar la aplicación de un factor de escala para hacer frente al cambio de escala introducido por el espejo. Se han tomado 6 imágenes y se han seleccionado para el test 7 puntos característicos situados dentro de una zona que no experimenta rotación en la imagen a lo largo de la secuencia. Se han considerado dos casos: aumento de la escala y disminución de la escala.

Para el caso de disminución de la escala se han extraído los puntos en la imagen donde estos se encontraban más lejos del punto principal. Los resultados de este caso se muestran en la figura 5.6 para cada parche por separado y en la figura D.6 para todos los parches.

Como puede apreciarse en todas las gráficas de la figura 5.6, los valores obtenidos para la correlación tanto en el punto exacto como para el punto con mejor correlación son sensiblemente superiores utilizando un factor de escala. Además, como puede observarse en la tercera gráfica, en las últimas imágenes, donde el cambio de escala respecto al parche inicial es mayor, usando los parches a los que no se les aplica el factor de escala el error de localización del punto es bastante elevado, mientras que usando el factor de escala éste es mínimo.

Para el segundo caso (aumento de la escala) se ha realizado la extracción de puntos en la imagen donde estos se encontraban más cerca del punto principal. Los resultados de este caso se muestran en la figura D.7 para cada parche por separado y en la figura D.8 para todos los parches.

A la vista de la figura D.7, en el caso de un aumento de la escala no existe una ventaja clara entre usar el factor de escala y no usarlo, ya que para cada punto se

obtienen valores de correlación prácticamente iguales tanto en la localización exacta como para el mejor candidato. Además el error de localización es en líneas generales bastante elevado para ambos parches, lo que se puede apreciar sobre todo en las gráficas de los parches 4, 5 y 6, donde el parche es emparejado erróneamente desde la segunda imagen de la secuencia tanto aplicando el factor de escala como sin aplicarlo.

La explicación más plausible es que cuando el parche de un punto característico aumenta su escala en la imagen aparece nueva información que no era visible cuando tenía una escala menor y que, por tanto, no es posible extraer del parche con menor resolución.

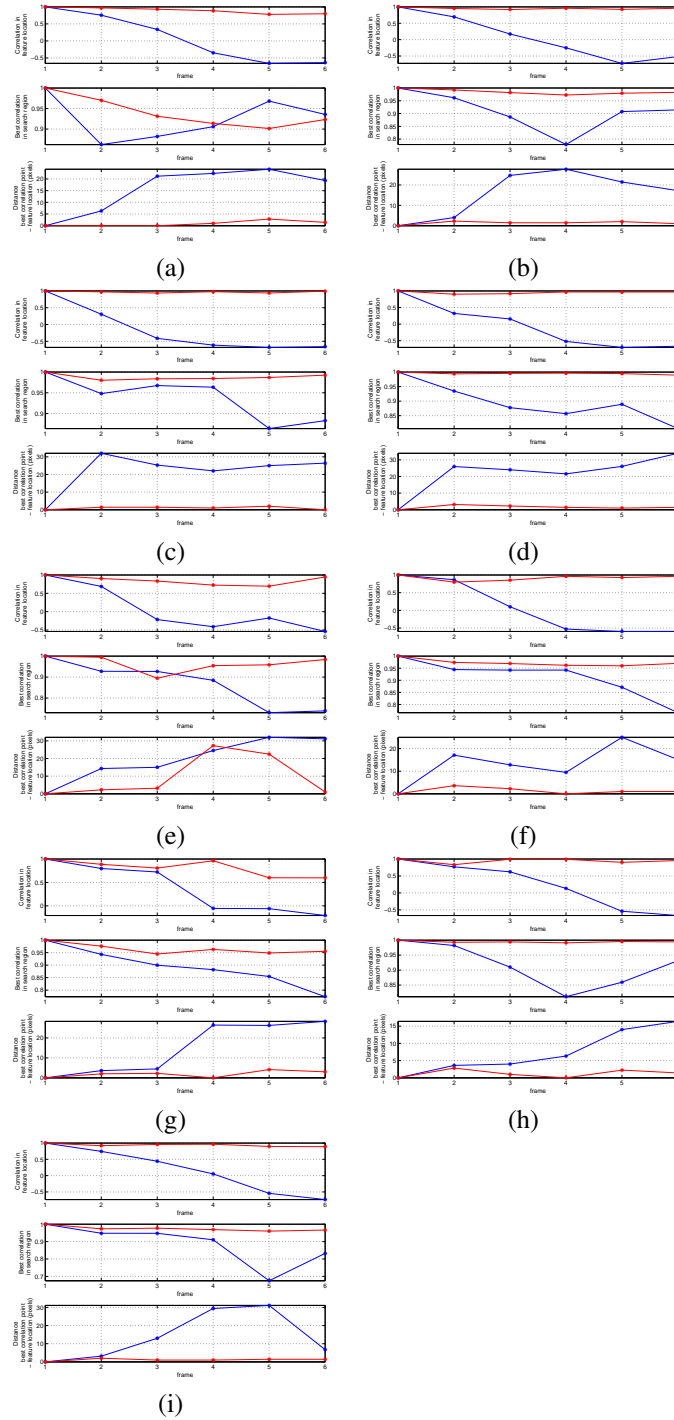


Figura D.1: Resultados de emparejamiento para los puntos seleccionados en el test 1. En rojo, con transformación de rotación. En azul, sin transformación de rotación.

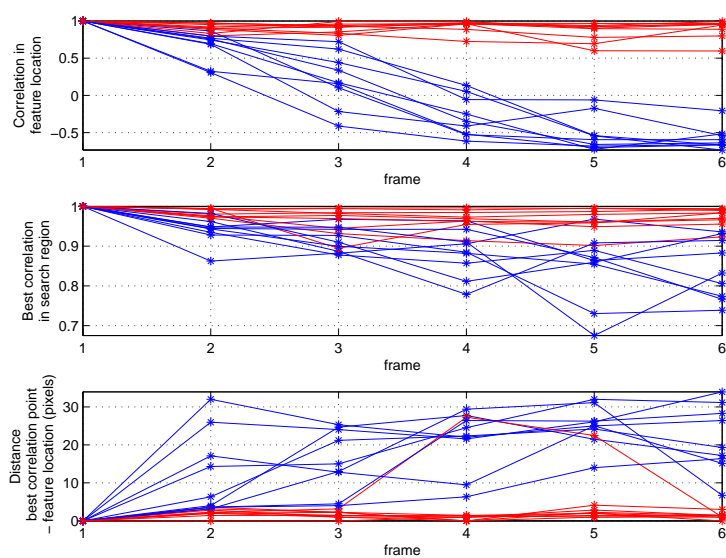


Figura D.2: Resultados de emparejamiento para todos los puntos del test 1. En rojo, con transformación de rotación. En azul, sin transformación de rotación.

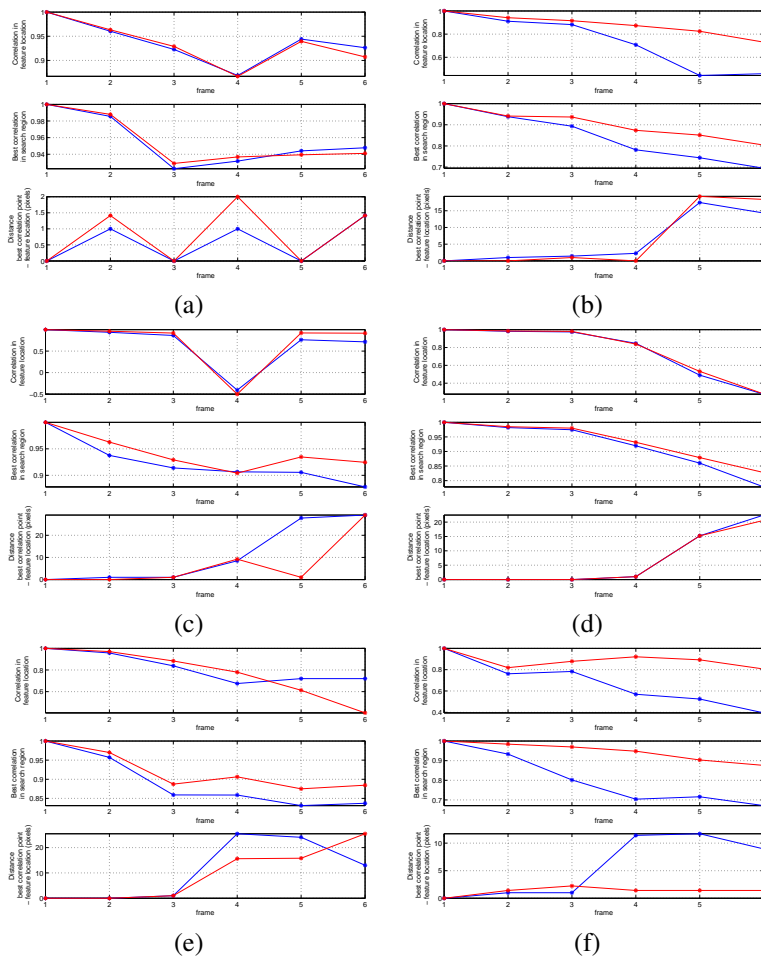


Figura D.3: Resultados de emparejamiento para los puntos seleccionados en el test 2. En rojo, con transformación de rotación. En azul, sin transformación de rotación.

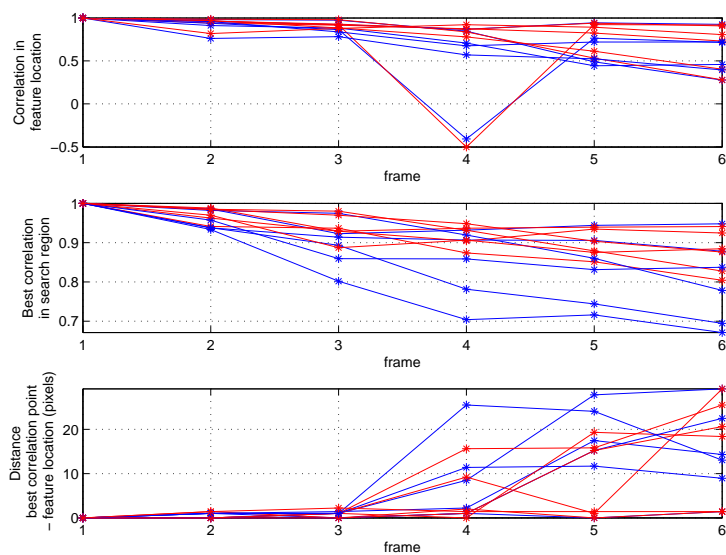


Figura D.4: Resultados de emparejamiento de todos los puntos del test 2. En rojo, con transformación de rotación. En azul, sin transformación de rotación.

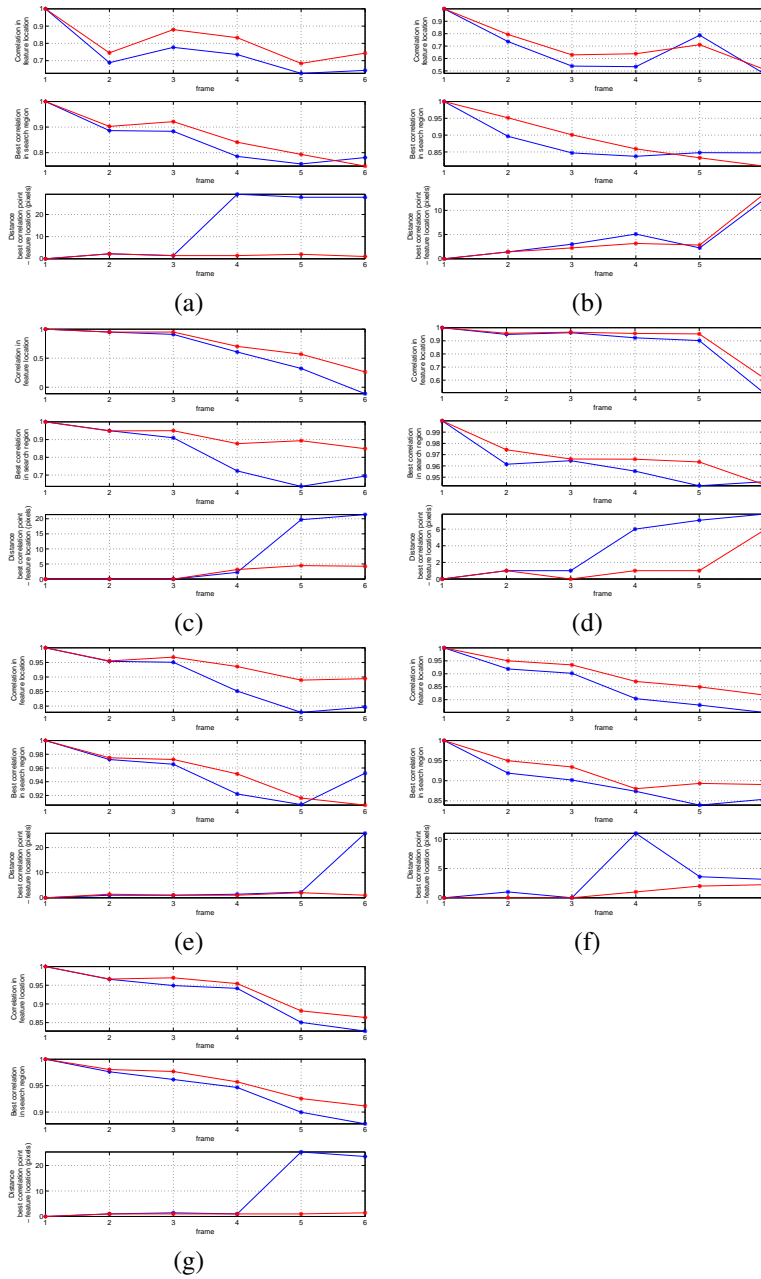


Figura D.5: Resultados de emparejamiento para los puntos seleccionados en el primer caso del test 3 (disminución de la escala). En rojo, con transformación de rotación. En azul, sin transformación de rotación.



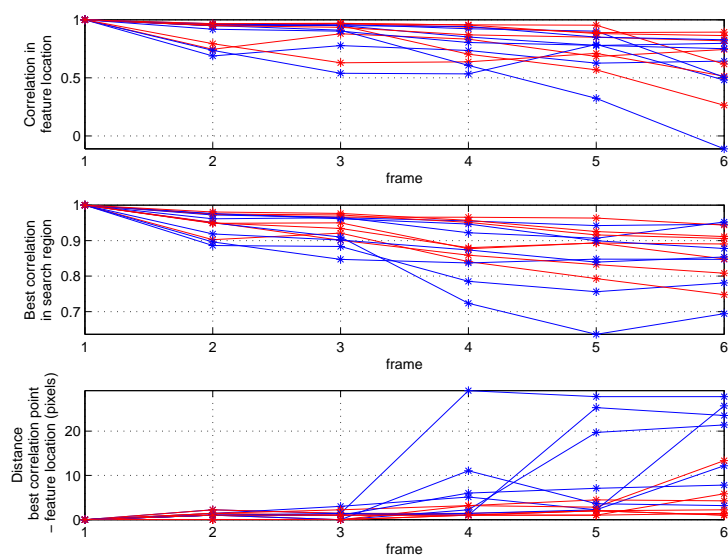


Figura D.6: Resultados de emparejamiento de todos los puntos para el primer caso del test 3 (disminución de la escala). En rojo, con transformación de rotación. En azul, sin transformación de rotación.

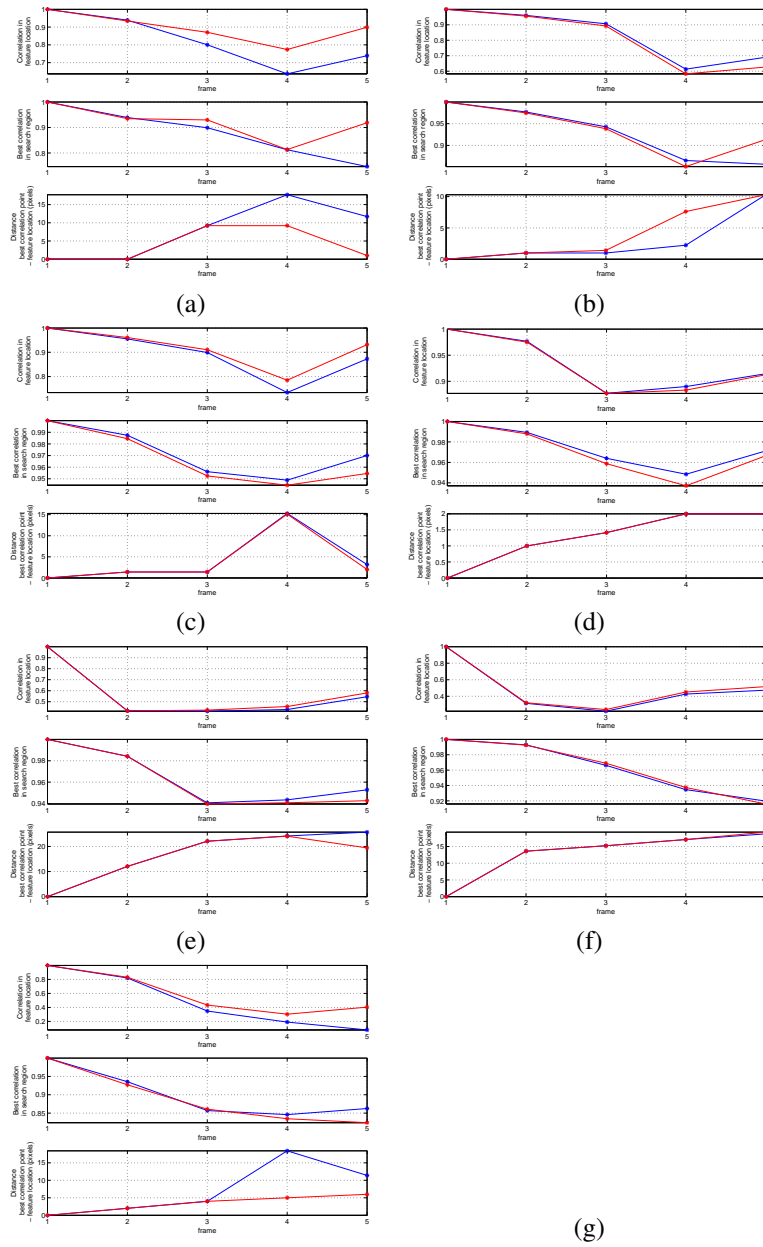


Figura D.7: Resultados de emparejamiento para los puntos seleccionados en el caso del test 3 (aumento de escala). En rojo, con transformación de rotación. En azul, sin transformación de rotación.

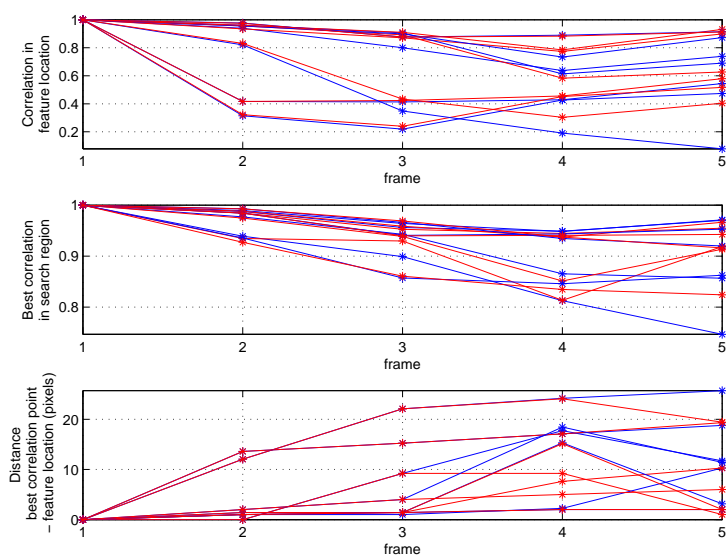


Figura D.8: Resultados de emparejamiento de todos los puntos para el segundo caso del test 3 (disminución de la escala). En rojo, con transformación de rotación. En azul, sin transformación de rotación.

---

## Anexo E

# Adapting a Real-Time Monocular Visual SLAM from Conventional to Omnidirectional Cameras

### Abstract

The SLAM (Simultaneous Localization and Mapping) problem is one of the essential challenges for the current robotics. Our main objective in this work is to develop a real-time visual SLAM system using monocular omnidirectional vision. Our approach is based on the Extended Kalman Filter (EKF). We use the Spherical Camera Model to obtain geometric information from the images which is valid for both types of cameras. To integrate this model in the EKF-based SLAM we linearize the direct and the inverse projections. We also introduce a new patch descriptor for catadioptric omnidirectional cameras which aims to be rotation and scale invariant. We perform experiments with omnidirectional images using our new patch and the conventional one. The experimentation confirms that our patch works better with omnidirectional cameras since features last longer and constructed maps are bigger.

### E.1. Introduction

An autonomous robot should be able to navigate through a previously unknown environment. The SLAM [31] problem tries to build a map of the surrounding and localize an autonomous robot relative to this map using only partial measurements of the environment. SLAM is usually formulated in a probabilistic way, i.e. the estimate of the robot position and map are computed as a probability distribution. Two main approaches are used for the computation of the probability distribution: the extended Kalman filter (EKF) [31] and the particle filter [2].

In Visual SLAM applications, image projections of relevant points known as local features are used as measurements. To extract and store the features on the image a feature extractor and a feature descriptor are used. The features (or corners) extractor process the image and detects the key-points on it. The image processing is a high

time-consuming step, which is critical for a real time application like SLAM. Focusing on this drawback, Rosten et al. [24] developed the feature extraction algorithm FAST (Features Accelerated Segment Test). They benchmark their FAST extractor with other widely used feature extractors showing that FAST extractor outperforms them in computational cost and in repeatability when viewing the scene from different positions. The descriptor provides an identifier to an extracted point so that it can be recognised in future measurements. The most basic descriptor is a patch of a certain size centered in the key-point, although there exists more sophisticated descriptors like SIFT [18], SURF [5], LBP [15], etc.

Since the seminal work of Davison [10], monocular SLAM has been a fertile research field. In this work we propose to combine state of the art robust EKF SLAM [8] with an omnidirectional sensor. Visual SLAM using omnidirectional cameras has been proposed by Scaramuzza et al in [25], where non-holonomic motion is assumed. In contrast our system assumes general camera motion.

Due to their 360° FOV, omnidirectional cameras features last longer on the image than in the case of conventional cameras. This is clearly noted in big camera rotations, where conventional camera loses most of its features due to its narrow FOV, while a omnidirectional camera can keep on tracking all the features [22]. The increased lifespan of the features on the image translates in a better estimation of the position of the features on the map, a lower need to initialise new features and a increased robustness.

However the omnidirectional images involve a more complex projection model, important image deformation, distortion and variable scale in the image. So, the feature descriptor should be modified for catadioptric cameras. In this way, Svoboda and Padjla [29] propose the use of patches with variable size and shape (active windows). Their experiments show that active windows provide best matching results than square windows. Ieng et al. [7] propose the computation of patches of different angular apertures for the same feature to overcome the matching problems derived from the varying resolution of the camera. Scaramuzza et al. [26] take advantage of the projection of vertical lines of the world as radial lines on the image. They propose a method to extract and match vertical lines with rotation invariant descriptors and apply this method to an EKF-SLAM. In [1] Andreasson et al. propose a modified SIFT feature with no scale invariance. To obtain rotation invariance they rotate each patch to the same global orientation. Lu and Zheng [19] combine the rotation invariant patch by Andreasson with a FAST extractor and a CS-LBP descriptor and they compare it with the SIFT algorithm.

Besides that, omnidirectional images require a more complex projection model to obtain geometric information from them. There are different projection models that describe central catadioptric systems [17], [27], [30], [33], however the most used is the Spherical Camera Model [12], [4]. This model can also be used with conventional cameras, which allows to develop a single application for both types of cameras. In [23] the integration of the Spherical Camera model in a monocular SLAM application is presented.

In this work we build on state of the art robust EKF monocular SLAM [8]. We introduce the Spherical Camera Model to deal with different kind of central omnidirectional and conventional cameras. Besides that we develop a patch for catadioptric cameras which considers rotation and scale invariance in function of mirror parameters. To reach rotation invariance we base on Andreasson proposal [1]. For scale invariance we will develop a formulation of the scale factor in function of the mirror parameters which can be applied in any kind of central camera and in particular in a

hiper-catadioptric system compound by a hiperbolic mirror coupled with a perspective camera.

The paper is structured as follows. Spherical Camera Model is described in Section 2. The SLAM problem is presented in Section 3 together with the Spherical Camera Model adaptation for the EKF. In section 4 our patch for the omnidirectional camera is formulated. Finally the results of the experiments with the new patch are presented in Section 5, and conclusions are presented in Section 6.

## E.2. The Spherical Camera Model

First of all we describe the projection model for the omnidirectional catadioptric systems. In [12] Geyer and Daniilidis present a unified model of projection for any catadioptric system with a single viewpoint. Barreto and Araujo extend this method in [4]. For the projection, we start with a 3D point expressed in homogeneous coordinates  $\mathbf{X} = [x, y, z, 1]$  The projection of this 3D point in the image is divided in three steps (see Fig. 1).

1) Point  $\mathbf{X}$  is mapped into a projective ray  $\mathbf{x}$  in the camera reference frame. This is done by  $\mathbf{P}$ , a conventional projection matrix  $\mathbf{x} = \mathbf{P}\mathbf{X}$ . If we assume the world and the mirror reference systems are the same,  $\mathbf{P} = [\mathbf{I} \mid \mathbf{0}]$ .

2) The ray  $\mathbf{x}$  in projected onto the unit sphere centered in the origin  $\mathbf{O}$ . The intersection point is projected to a virtual projection plane  $\pi$  through the virtual projection center  $\mathbf{C}_P = (0, 0, -\xi)^T$  yielding the point  $\mathbf{x}'$ . These two steps are coded by the non-linear function  $h$ :

$$\mathbf{x}' = h(\mathbf{x}) = \begin{pmatrix} x \\ y \\ z + \xi \sqrt{x^2 + y^2 + z^2} \end{pmatrix} \quad (\text{E.1})$$

3) The virtual plane  $\pi$  is transformed in the image plane  $\pi_{IM}$  through a homographic transformation  $H_c$

$$\mathbf{x}'' = H_c \mathbf{x}' \quad (\text{E.2})$$

$$H_c = K_c R M_c \quad (\text{E.3})$$

where  $K_c$  includes the camera parameters,  $M_c$  includes the mirror parameters [12] and  $R$  is the rotation matrix of the camera with respect to the mirror. By assuming a pin-hole camera model and  $R = \mathbf{I}$ , the transformation  $H_c$  yields:

$$H_c = \begin{bmatrix} \eta f & 0 & u_0 \\ 0 & \eta f & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \gamma & 0 & u_0 \\ 0 & \gamma & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{E.4})$$

where  $\gamma = \eta f$  is the generalized focal length of the camera-mirror system with  $\eta$  a mirror parameter and  $f$  the focal length of the camera.

4) Finally image coordinates are calculated by dividing  $\mathbf{x}''$  by its  $z''$  coordinate:

$$\mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix} = f_u(\mathbf{x}'') = \begin{pmatrix} \frac{x''}{z''} \\ \frac{y''}{z''} \end{pmatrix} \quad (\text{E.5})$$

The parameter of the model,  $\xi$  depends only on the system modeled and the geometry of the mirror. For conventional cameras  $\xi = 0$ .  $\xi = 1$  for catadioptric systems

with parabolic mirror and orthographic camera, and  $0 < \xi < 1$  with hyperbolic mirror and perspective camera.

With this model it is also possible to estimate the 3D ray from where the image point comes. That projection is named the inverse projection model. It starts with the point image coordinates  $\mathbf{u} = (u, v)^\top$ , being  $\mathbf{x}'' = (u, v, 1)^\top$ . The equations of the inverse projection model are:

$$\mathbf{x}' = \mathbf{H}_c^{-1} \mathbf{x}'' \quad (\text{E.6})$$

$$\mathbf{x} = \mathbf{h}^{-1}(\mathbf{x}') = \begin{pmatrix} x' \\ y' \\ z' - \frac{\xi(x'^2 + y'^2 + z'^2)}{\xi z'^2 + \chi} \end{pmatrix} \quad (\text{E.7})$$

$$\text{where } \chi = \sqrt{(1 - \xi^2)(x'^2 + y'^2 + z'^2)}$$

### E.3. Simultaneous Localisation And Mapping

The most used SLAM algorithms are based on the Kalman Filter, a filter that predicts the state of linear systems. As the geometry imposes non-linear relations, the Extended Kalman Filter (EKF) [31] is used. The EKF linearize the non-linear functions by approximating them to its first order Taylor series. The EKF is divided into two parts. In the first part, *Prediction*, the new state of the system is estimated from the previous time step state through the motion model. The second part of the algorithm, *Update*, uses the measurements of the environment to improve the new state prediction. The full state vector, composed of both the map and last camera location, is modelled as multidimensional Gaussian distribution coded by its mean vector and covariance matrix.

The state of the system is given by the state vector  $\mathbf{x}$

$$\mathbf{x} = (\underbrace{\mathbf{r}, \mathbf{q}, \mathbf{V}, \omega}_{\text{Camera state}}, \underbrace{x_i, y_i, z_i, \theta_i, \phi_i, \rho_i, \dots}_{\text{3D points (IDP)}}, \dots, \underbrace{X_{i+n}, Y_{i+n}, Z_{i+n}, \dots}_{\text{3D points (XYZ)}}) \quad (\text{E.8})$$

where  $\mathbf{r}_{(3 \times 1)}$  is the camera pose,  $\mathbf{q}_{(4 \times 1)}$  is the quaternion of its orientation and  $\mathbf{V}_{(3 \times 1)}$  and  $\omega_{(3 \times 1)}$  are its linear and angular velocities, respectively. The state size of the map features depends on the depth uncertainty they have. Features with large depth uncertainty are parameterised in inverse depth parametrisation (IDP) [9]. This parametrization is used for recently initialised features. They are initialised with an arbitrary depth prior of  $\rho_{0i}$  with large uncertainty. In successive observations of the feature, depth estimation is gradually refined. If the depth uncertainty of a feature decreases under a certain threshold then the state of the feature is given by its cartesian coordinates in the world reference frame. Since  $\mathbf{x}$  has  $n$  dimensions the state covariance matrix  $\mathbf{P}$  is a squared  $n \times n$  matrix.

#### E.3.1. The Spherical Camera Model for the EKF

The EKF algorithm requires the first derivative of the measurement equation. In order to adapt the Spherical Camera Model to the EKF algorithm we have to compute the model derivatives, for the direct and the inverse projection [23].

So, the jacobian of the Spherical Camera Model projection is

$$\mathbf{J} = \mathbf{J}_{f_u} \mathbf{H}_C \mathbf{J}_h \quad (\text{E.9})$$



$$\mathbf{J}_{fu} = \begin{bmatrix} \frac{1}{z''} & 0 & -\frac{x''}{z''^2} \\ 0 & \frac{1}{z''} & -\frac{y''}{z''^2} \end{bmatrix} \quad (\text{E.10})$$

$$\mathbf{J}_h = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{\xi x}{\rho} & \frac{\xi y}{\rho} & 1 + \frac{\xi z}{\rho} \end{bmatrix} \quad (\text{E.11})$$

where  $\rho = \sqrt{x^2 + y^2 + z^2}$

To initialize new features we require a way to estimate the covariance of the new feature from the covariance of the image point. This estimation can be done through the Jacobian of the Spherical Camera Model inverse projection

$$\mathbf{J}^{-1} = \mathbf{J}_{h-1} \mathbf{H}_C^{-1} \quad (\text{E.12})$$

$$\mathbf{J}_{h-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{\xi x'}{\chi} & -\frac{\xi y'}{\chi} & 1 - \frac{\xi(z' - \xi \frac{x'^2 + y'^2 + z'^2}{\xi z' + \chi})}{\chi} \end{bmatrix} \quad (\text{E.13})$$

where  $\chi = \sqrt{(1 - \xi^2)(x'^2 + y'^2 + z'^2)}$

### E.3.2. Data Association and Map Management

Robust 1-point-RANSAC [8] based active search is applied. At each prediction step, an elliptical search region is computed from the measurement prediction and its corresponding covariance innovation. Correlation is computed for every pixel inside the search region. The pixel scoring highest is selected as putative match. In a second stage, joint scene rigidity is checked for all the putative matches and spurious matches are detected. Active search is both efficient, because only a reduced fraction of the image is searched, and robust because of the reduced false positive rate in the putative matching computation. New point features are initialized when the number matched map points are under a threshold. To improve geometrical condition map features have to be spread all over the image. FAST key-points scoring higher are searched in feature depleted image areas. From these key-points new map features are initialized. In order to keep complexity low, map features that are repeatedly not detected are marginalized out from the state vector.

## E.4. New patch formulation

In this section we develop the formulation of a patch for catadioptric cameras invariant to rotation and scale.

### E.4.1. Rotation invariance

For the rotation invariance we adopted the idea proposed by Andreasson et al. [1]. The idea consists in extracting a squared oriented patch in the radial direction from the principal point to the feature by bilinear interpolation and storing as feature descriptor the patch rotated to a fixed orientation. However as described in Section 3, in the used SLAM application, matching is done by looking for the patch in a uncertainty region, rather than extracting all the features and comparing them with the existing ones. When trying to correlate each pixel inside the search region, each candidate patch should be

determined by bilinear interpolation in the non-natural radial and polar directions of the image, which would be time consuming.

To solve this, we combine this idea with the implementation existing in the SLAM application. In this application a bigger patch is extracted together with the descriptor patch during feature initialisation. Before the matching process, this big patch is warped by an homographic transformation which tries to predict how the appearance patch varies depending of the variation of the position of the camera respect to the position in which the feature was initialised. New patch for correlation is extracted from the center of the warped big patch. This way patches for correlation are always determined in the horizontal and vertical directions of the image and bilinear interpolation is only computed during the big patch warping.

For its use with omnidirectional cameras, we transform the patch by a rotation transformation given by the variation of the polar angle in the image of the initial point and its prediction  $\Delta\theta$ .

### E.4.2. Scale invariance

To reach scale invariance, we develop the simple idea of scaling the patch by a given scale factor. To consider the variable resolution in the catadioptric image, a theoretical formula was obtained in function of the mirror parameters and the image position.

To obtain it, first we define a point in the 3D space in homogeneous coordinates at a distance or depth  $D$  from the camera with an azimuth  $\phi$  and an elevation of  $\theta$ . Due to the rotational symmetry of the mirror and for the sake of simplicity an azimuth angle of  $\phi = 0$  is taken without loss of generality. Therefore the coordinates of the 3D point yield  $\mathbf{X}_0 = (D \cos \theta, 0, D \sin \theta, 1)^\top$ . According to the spherical camera model this point is projected on the image plane as  $\mathbf{p}_0 = (\frac{\gamma \cos \theta}{\xi + \sin \theta}, 0, 1)^\top$ . To express the coordinates in the image in the reference frame attached to the principal point we made  $u_0 = v_0 = 0$  in the matrix  $H_C$ , which holds for the rest of the formulation. The norm of the projected point is the distance from the principal point  $R_{im}$

$$R_{im} = \|\mathbf{p}_0\| = \frac{\gamma \cos \theta}{\xi + \sin \theta} \quad (\text{E.14})$$

Now let us take a sphere of radius  $r$  centered on the previously projected 3D point. Under the assumption that  $r \ll D$ , points on the surface of the sphere can be projected through the Jacobian computed in its centre:

$$J_{p=p_0} = J_{fu} H_c J_h P \quad (\text{E.15})$$

After the calculations and some algebraic manipulation the following expression for the Jacobian results:

$$J_{m=m_0} = \frac{\gamma}{D(\xi + S_\theta)^2} \begin{bmatrix} S_\theta(1 + \xi S_\theta) & 0 & -C_\theta(1 + \xi S_\theta) \\ 0 & \xi + S_\theta & 0 \end{bmatrix} \quad (\text{E.16})$$

where  $S_\theta$  and  $C_\theta$  are  $\sin \theta$  and  $\cos \theta$  respectively

Projecting the sphere we have an ellipse in the image. Assuming  $r \ll D$  we can obtain after some manipulation the expression of the image ellipse.

The semi-major axis ( $r_{im}^+$ ) and semi-minor axis ( $r_{im}^-$ ) yield:

$$r_{im}^+ = \gamma \frac{r}{D} \frac{1 + \xi \sin \theta}{(\xi + \sin \theta)^2} \quad (\text{E.17})$$

$$r_{im}^- = \gamma \frac{r}{D} \frac{\xi + \sin \theta}{(\xi + \sin \theta)^2} \quad (\text{E.18})$$

If we solve (E.14) for  $\theta$  we can put  $\sin \theta$  in the previous expressions as a function of  $R_{im}$ . After some substitutions and manipulation we obtain a second order equation with unknown  $\sin \theta$ . By solving the equation and selecting the solution with physical meaning, we obtain:

$$\sin \theta = f\left(\xi, \frac{R_{im}}{\gamma}\right) = \frac{\sqrt{1 + \left(\frac{R_{im}}{\gamma}\right)^2(1 - \xi^2)} - \xi\left(\frac{R_{im}}{\gamma}\right)^2}{1 + \left(\frac{R_{im}}{\gamma}\right)^2} \quad (\text{E.19})$$

According to these formulas the size of an object on the image depends on the following parameters:

- Real size of the object ( $r$ )
- Distance of the object to the camera ( $D$ )
- Camera-mirror parameters  $\xi$  and  $\gamma$
- Distance in the image to the principal point ( $R_{im}$ )

Note that the real size of the object is not relevant since it does not change. The depth of the object in the scene  $D$  is the most problematic contribution since it is not observable in one image. As explained in Section 3, during SLAM execution a new detected feature is initialized in IDP with an arbitrary depth value with a quite high uncertainty which is not reliable to calculate the scale factor. However, in the following steps this depth value is refined and when its uncertainty reaches a reasonable value the feature is fully initialised and parameterised by cartesian coordinates. So, in the implementation of the new patch we will only consider the scale factor when matching fully initialised features.

Concerning the contribution of the distance to the image center  $R_{im}$ , we must choose between two expressions depending on which semiaxis of the projected sphere is used to calculate the size. At this point, it must be noted that radial distortion was not taken into account in the projection of the sphere to simplify the formulation. This would make the formula calculated with the major semiaxis less accurate since this semiaxis is determined in the radial direction. To prove this, a simulation has been launched using a catadioptric camera model with distortion parameters and projecting a set of spheres with a shift on elevation angle and at a constant distance from the centre of projection. The results have been compared with the prediction of the theoretical formulas (Fig. E.1) and proved that the minor semiaxis is more suitable to calculate the scale factor.

So, the final expression to calculate the scale factor remains:

$$k = \frac{r_{im2}^-}{r_{im1}^-} = \frac{D_2 \xi + f\left(\xi, \frac{R_{im2}}{\gamma}\right)}{D_1 \xi + f\left(\xi, \frac{R_{im1}}{\gamma}\right)} \quad (\text{E.20})$$

with

$$f\left(\xi, \frac{R_{im_i}}{\gamma}\right) = \frac{\sqrt{1 + \left(\frac{R_{im_i}}{\gamma}\right)^2(1 - \xi^2)} - \xi\left(\frac{R_{im_i}}{\gamma}\right)^2}{1 + \left(\frac{R_{im_i}}{\gamma}\right)^2} \quad (\text{E.21})$$

### E.4.3. Computation of patch transformation

Once the methods to calculate the rotation transformation and the scale have been provided we compute the patch warping. If we define  $h_{BP}$  and  $h_P$  as the half of the

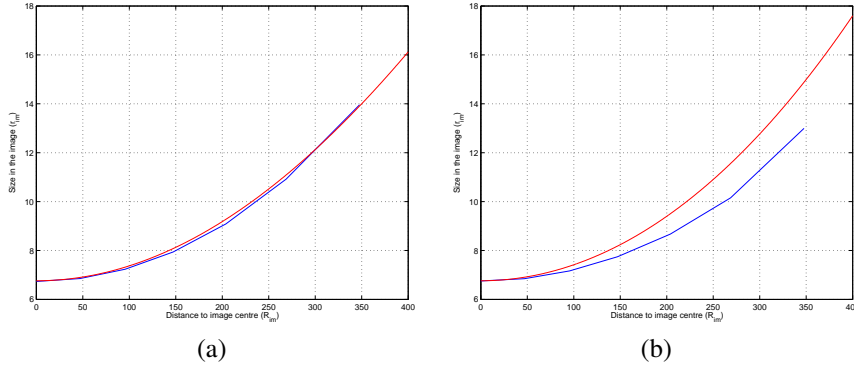


Figure E.1: Comparison of the theoretical formulas to calculate the ellipse semiaxis in which the sphere is projected (red) with the results of a simulation using a camera model with radial distortion (blue) The prediction for the minor semiaxis (a) is more accurate than the one for the major semiaxis(b).

sizes of the big patch and the descriptor patch respectively, the computation of the warping is done in the following steps:

1) Check the condition  $k > \sqrt{2} \frac{h_P}{h_{BP}} \cos(\frac{\pi}{4} - \Delta\theta) + 0,1$  to avoid extracting a patch with no information from the big patch. If  $k$  does not fill this condition it is set to the threshold value.

2) Calculation of the transformation matrix:

$$H = H_{tr} H_S H_{tr}^{-1} \quad (E.22)$$

$$H_S = \begin{bmatrix} k \cos(\Delta\theta) & -k \sin(\Delta\theta) & 0 \\ k \sin(\Delta\theta) & k \cos(\Delta\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (E.23)$$

with  $H_S$  the isotropic transformation matrix which combines the rotation transformation  $R_{\Delta\theta}$  with the scale factor  $k$  and  $H_{tr}$  is a translation matrix to translate the patch coordinate frame to the center of the patch.

3) Computation of the warped patch by doing the inverse mapping to the original big patch and performing a bilinear interpolation.

$$X_{BP} = H_S^{-1} X_{WP} \quad (E.24)$$

4) Extraction of the new descriptor from the center of the warped patch.

## E.5. Experiments

In this section the experiments carried out to evaluate the omnidirectional visual SLAM and the new patch are presented. The first experiment consists of three tests to evaluate separately the rotation and the scale invariance of the new patch by comparison with the normal patch. In the second experiment we evaluate the performance of the new patch during SLAM.

The images used to lead the experiments were taken from one of the image databases provided by The Rawseeds Project <sup>1</sup>. This database consists of a sequence

<sup>1</sup>[HTTP://www.rawseeds.org](http://www.rawseeds.org)

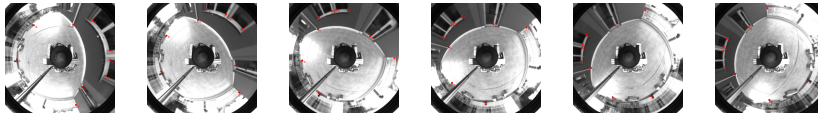


Figura E.2: Image sequence taken for test 1 (180° rotation). Selected corners for matching are shown in red

with more than 32000 frames acquired by a robot equipped with a hyper-catadioptric camera.

### E.5.1. Experiment 1

In the first experiment we have carried out three tests where we have decoupled the matching process from the SLAM algorithm to make a preliminary evaluation of our new patch attributes. In the first and the second test we evaluate an only rotation invariant patch under a rotation of 180° and a translation respectively. In the third test we evaluate the performance of an only scale invariant patch. The set up of the three test was quite similar. First, we extracted corners on the first frame with the FAST extractor. Among the extracted corners, we selected some features and stored their locations and their patches.

Like the matching process has been decoupled from the SLAM algorithm the estimated locations of the features were manually selected on each frame (Fig. E.2) and the search region was fixed to a 50x50 pixels square. The matching in the selected frames is done by obtaining for each feature the best correlation inside the search region, which is the method used in the SLAM application where the tested patches are going to be applied.

For each feature and frame, we have defined the following variables to be measured:

- Correlation in estimated feature location
- Best correlation in search region
- Distance between estimated feature location and best correlation location

#### Test 1: Rotated patch and 180° rotation

For this test a sequence in which the robot rotates 180° was selected. From this sequence we have extracted 6 frames which are spaced by 20 frames between them. We have selected 9 features to carry out the test(Fig. E.2).

The results show that rotated patches provide a better correlation value on the true feature location, which confirms that they are more rotation invariant than the non rotated patches (Fig. E.3). The rotated patch also provides by far better values for the best correlation inside the search region (all of them above 0,9) as well as a very low distance between true feature location and matched location.

#### Test 2: Rotated patch and translation

For the second test a sequence containing only camera translation was taken. 6 frames were extracted with intervals of 20 frames between them and the number of selected features was 6 (Fig. E.4).

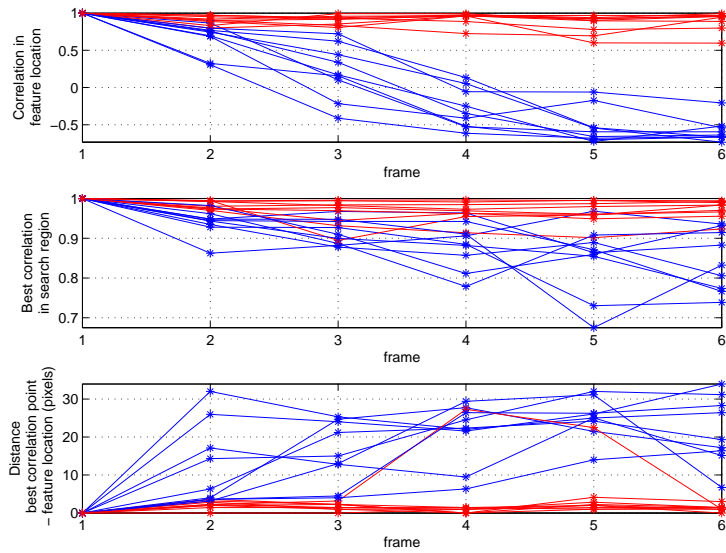


Figura E.3: Matching results of test 1. In red, with oriented patch. In blue, with not-oriented patch

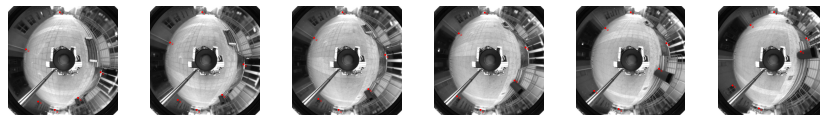


Figura E.4: Image sequence taken for test 2 (translation). Selected corners for matching are shown in red

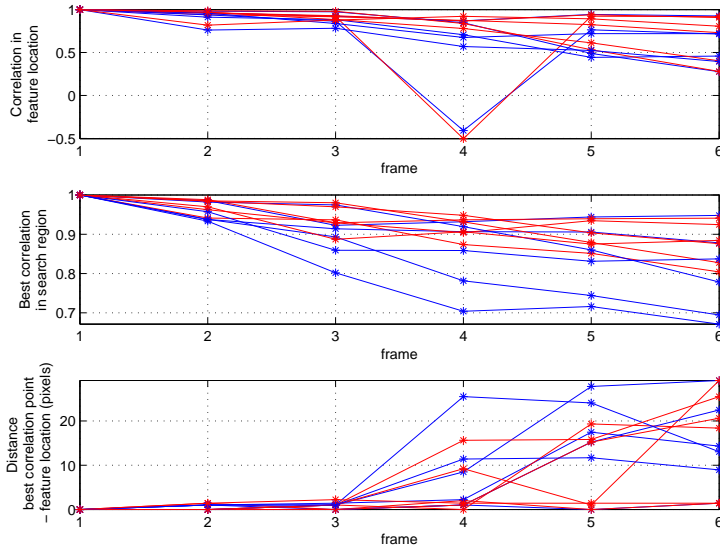


Figura E.5: Matching results of test 2. In red, with oriented patch. In blue, with not-oriented patch

The results (Fig. E.5) reveal that under camera translation, slightly better correlation values are obtained using a rotated patch. It can be seen too, that as the distance the robot has translated increases, the matchings tend to be made in the wrong location for both patches. One possible reason is that as the robot translates the features change their scale and their point of view and can become hidden by other scene objects.

### Test 3: Scaled patch

In this test we have evaluated the performance of the matching process with respect to the scale changes. As this experiment is decoupled from SLAM, it is not possible to determine the depth of the patches extracted. Therefore the contribution of the depth to the scale factor is not considered. So, an image acquisition without depth changes of features has been made following the next steps:

- Select a zone in the scene with potential patch richness and situated far enough from the camera so that  $D \rightarrow \infty$ .
- Capture images while camera rotates so that the selected zone moves only along the radial direction. Like infinite distance has been assumed, little camera displacements during capture are not problematic.

A sequence of 6 images was taken for the test. The number of selected features was 7. To evaluate the performance of the patches under scale change, the features were selected in a zone with no orientation change in the image (Fig. E.6). Two cases have been carried out to prove the performance under scale decrease  $k < 1$  and scale increase  $k > 1$ . In the scale decrease case, the extraction of the features was made in the image where the zone of extraction was the furthest from the image centre. For the scale increase case the order of the images has been inverted.

The results of the tests show that in scale decrease (Fig. E.7) the patch with scaling offers a better performance than a normal patch while in the case of scale



Figure E.6: Image sequence taken for test 3 (scale change). Selected corners for matching are shown in red

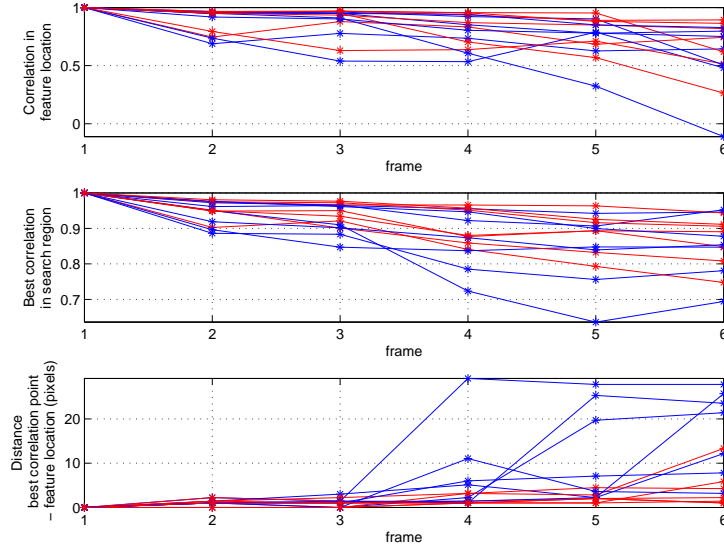


Figure E.7: Matching results of test 3 (scale decrease). In red, with scaled patch. In blue, with not-scaled patch

increase(Fig.E.8) both patches perform in a similar way.

### E.5.2. Experiment 2

In this experiment we have evaluated our visual SLAM approach for omnidirectional cameras with the Real-Time application developed by Davison et al. . For the evaluation we selected a long outdoor sequence from the database provided by the Rawseeds Project.

To compare our new patch with the normal patch we have run the sequence using both patches with different correlation thresholds for matching and we have measured three variables ratios:

- Total number of features initialised ( $FI$ ) .
- Matchings per feature ratio: ( $R_m = \frac{Total\ matchings}{FI}$ ) .
- Features in map per feature initialised ratio: ( $R_f = \frac{Final\ map\ size}{FI}$ )

The results in Table E.1 show that the patch for omnidirectional cameras (warped patch) performs better than the patch for conventional cameras with no warping (normal patch), as it initialises less features and is capable of obtaining more information for SLAM per initialised feature. However the table also shows that the



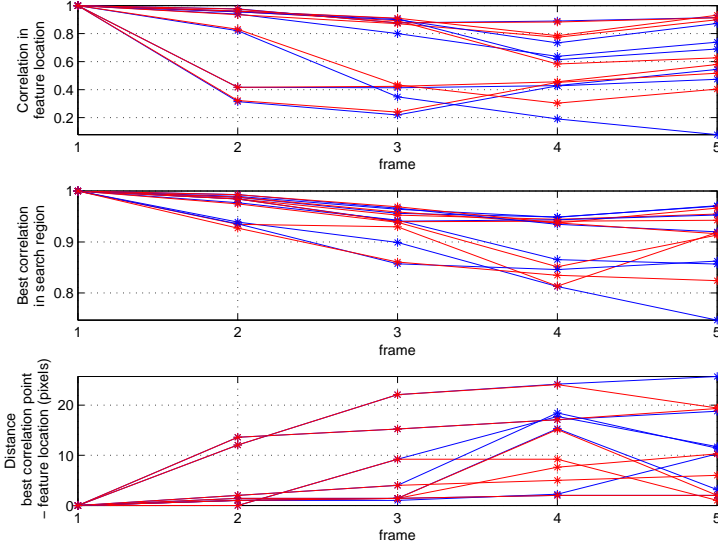


Figura E.8: Matching results of test 3 (scale increase). In red, with scaled patch. In blue, with unscalled patch.

Tabla E.1: Matchings and features in map per initialised feature

Correlation threshold	Warped patch			Normal patch		
	$FI$	$R_m$	$R_f$	$FI$	$R_m$	$R_f$
0,8	8648	22,31	0,1	8923	19,75	0,087
0,9	9834	19,38	0,062	10854	16,09	0,046
0,95	13189	13,31	0,027	14970	10,56	0,019

correlation threshold has more influence on the measured variables than the kind of patch used. In Fig. E.9 the projections on the XY plane and the YZ plane of the trajectory obtained with the new patch and a correlation threshold of 0,8 are shown. Note that although the MonoSLAM application estimates 3D camera motion, being not bounded to a 2D plane, the obtained trajectory shows that the camera is moving on the ground plane.

Finally we compared this trajectory with respect to the ground truth provided by the GPS data. As scale is not observable by one single camera, for the comparison we scaled the trajectory and aligned it with the trajectory obtained with the GPS (Fig. E.10). To evaluate the accuracy of the SLAM trajectory numerically we have calculated the mean error of the distance between the corresponding points of both trajectories. The calculated mean error is  $\mu_{err} = 3,44 m$  with a standard deviation of  $\sigma = 1,93 m$  and a maximum error of  $max_{err} = 6,73 m$ . If we divide the mean error and the trajectory lenght, we obtain a relative mean error of 1 %.

## E.6. Conclusion

In this work we have developed a Visual SLAM algorithm for omnidirectional cameras building on state of the art EKF monocular SLAM [8] for conventional

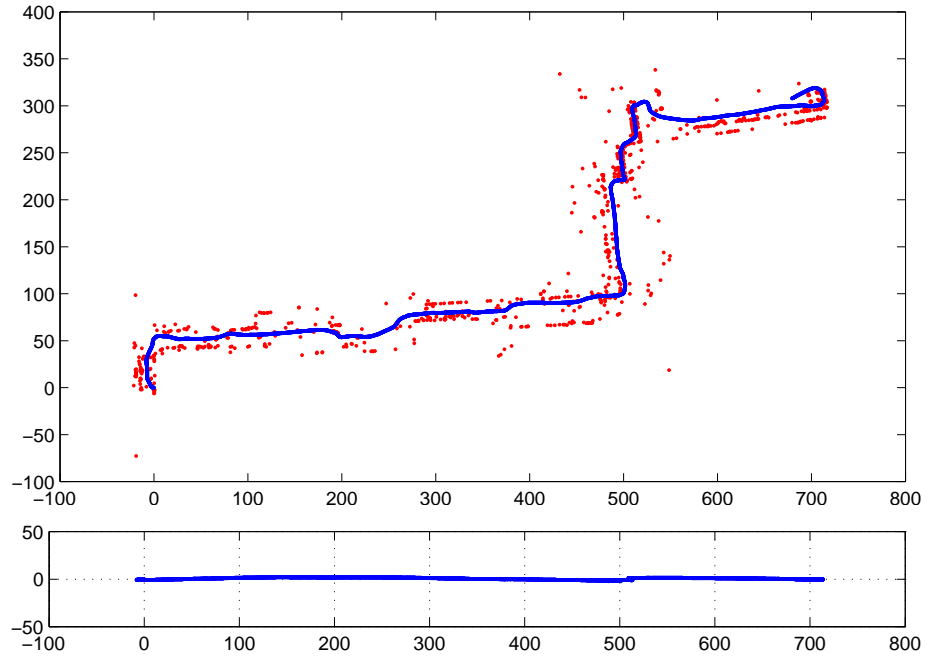


Figura E.9: SLAM trajectory with correlation threshold 0,8 using the warped patch projected on the XY plane (up) and on the YZ plane (down) The red dots are the map features



Figura E.10: GPS trajectory (red) and SLAM trajectory (green) superposed on the satellite image of the Campus of Bovisa (Milan) where the sequences were acquired

## ANEXO E. ADAPTING A REAL-TIME MONOCULAR VISUAL SLAM FROM CONVENTIONAL TO OMNIDIRECTIONAL CAMERAS

---

cameras. Two main modifications have been made: the implementation of the Spherical Camera Model for projection and the formulation of a new patch for omnidirectional cameras which is rotation and scale invariant. Then we have lead experiments to compare the new patch with the conventional patch. First we have tested the matching process decoupled from the SLAM. Once the supperiority of our new patch has been proven we have run the SLAM algorithm in a 340 meters long trajectory. Results have shown that the obtained trajectory estimation is quite accurate, which encourages us to make experiments with longer trajectories in the future.



# Índice de figuras

2.1. Proyección de un punto 3D $X_w$ en el plano imagen utilizando el Modelo de la Esfera . . . . .	12
2.2. Pasos de la proyección del Modelo de la Esfera . . . . .	13
3.1. Aplicación de SLAM. . . . .	16
4.1. Los puntos de la escena que estén contenidos dentro del cono de la figura no se pueden proyectar en el plano imagen. . . . .	23
4.2. Máscara que se aplica a las cámaras omnidireccionales para no inicializar ni emparejar puntos que permanecen inmóviles en la imagen por ser solidarios a la cámara. . . . .	23
4.3. El parche de un punto característico recién inicializado, se rota a una orientación fija antes de ser guardado como descriptor para futuros emparejamientos. De esta forma el parche descriptor es invariante frente a rotación de la cámara. . . . .	25
4.4. Comparación de las fórmulas teóricas para calcular los semiejes de la elipse en la que la esfera es proyectada (rojo) con las gráficas obtenidas de una simulación con el modelo de una cámara con distorsión radial (azul). La gráfica (a) corresponde al semieje menor y la gráfica (b) al semieje mayor. . . . .	27
4.5. Error absoluto (arriba) y relativo (abajo) en función de la distancia al punto principal entre las longitudes de los semiejes obtenidas por simulación y las obtenidas por la formulación teórica. En azul, para el semieje mayor y en rojo para el semieje menor. . . . .	28
4.6. Evolución del tamaño de un parche en la imagen ( $r_{im}$ ) en función de su distancia al punto principal ( $R_{im}$ ) y de la distancia en metros ( $z$ ) del plano horizontal en el que se encuentra el parche en la escena respecto del plano en el que se mueve la cámara. Si el parche se encuentra por debajo de la cámara ( $z < 0$ ), $r_{im}$ decrece con $R_{im}$ , mientras que si se encuentra por encima $r_{im}$ aumenta con $R_{im}$ . . . . .	29
5.1. Secuencia de imágenes tomada para el test 1 (Rotación de la cámara de $180^\circ$ ). Los puntos seleccionados para emparejamiento están marcados en rojo . . . . .	33
5.2. Resultados de emparejamiento para el test 1. En rojo, con transformación de rotación. En azul, sin transformación de rotación. . . . .	33
5.3. Secuencia de imágenes tomada para el test 2 (Traslación). Los puntos seleccionados para emparejamiento están marcados en rojo . . . . .	35

5.4.	Resultados de emparejamiento para el test 2. En rojo, con transformación de rotación. En azul, sin transformación de rotación. . . . .	35
5.5.	Secuencia de imágenes tomada para el 3 (Cambio de escala). Los puntos seleccionados para emparejamiento están marcados en rojo . .	36
5.6.	Resultados de emparejamiento para el test 3 (disminución de la escala). En rojo, con parche escalado. En azul, con parche sin escalar. . . . .	37
5.7.	Resultados de emparejamiento para el test 3 (aumento de la escala). En rojo, con parche escalado. En azul, con parche sin escalar. . . . .	37
5.8.	Trayectoria SLAM, obtenida con un umbral de correlación de 0,8 y usando el parche para cámara omnidireccional, proyectada en el plano XY (arriba) y en el plano XZ (abajo). Los puntos rojos corresponden a los puntos característicos en el mapa final. La figura sólo da una idea cualitativa de la trayectoria ya que las dimensiones dependen de una escala estimada de forma arbitraria, al no ser observable. . . . .	39
5.9.	Trayectoria GPS (rojo) y trayectoria SLAM (verde) superpuestas sobre la imagen de satélite del Campus de Bovisa (Milán) donde fueron tomadas las secuencias por el equipo de Rawseeds . . . . .	40
5.10.	Orientación de la cámara obtenida del SLAM (azul) y de los datos de posición del GPS (rojo). El ruido en esta última se debe a que ha sido obtenida por diferenciación. . . . .	40
5.11.	Trayectoria SLAM (secuencia corta con casco) proyectada en el plano XY (arriba) y en el plano XZ (abajo). Los puntos rojos corresponden a los puntos característicos en el mapa final. Las dimensiones dependen de una escala estimada de forma arbitraria, al no ser observable. . . .	42
5.12.	Trayectoria GPS (rojo) y trayectoria SLAM (verde) de la secuencia corta superpuestas sobre la imagen de satélite de la zona donde se hizo la adquisición. Dada la imprecisión del GPS a partir del punto marcado en negro en ambas trayectorias, se ha utilizado solamente el tramo hasta este punto para realizar el escalado y alineado de la trayectoria SLAM y el posterior análisis cuantitativo de la precisión. . . . .	43
5.13.	Orientación de la cámara obtenida del SLAM (azul) y de los datos de posición del GPS (rojo). Existe ruido en esta última ya que ha sido obtenida por diferenciación a partir de la posición dada por el GPS. . .	44
5.14.	Trayectoria SLAM (secuencia larga con casco) proyectada en el plano XY (arriba) y en el plano XZ (abajo). Los puntos rojos corresponden a los puntos característicos en el mapa final . . . . .	46
5.15.	Trayectoria GPS (rojo) y trayectoria SLAM (verde) de la secuencia larga superpuestas sobre la imagen de satélite de la zona donde se hizo la adquisición. El escalado de la trayectoria se ha realizado con el tramo desde el inicio hasta el punto marcado en negro en ambas trayectorias.	47
5.16.	Orientación GPS (rojo) y orientación SLAM (verde) de la cámara durante la secuencia larga. . . . .	47
B.1.	Robot Robocom con el que se han adquirido los datos . . . . .	58
D.1.	Resultados de emparejamiento para los puntos seleccionados en el test 1. En rojo, con transformación de rotación. En azul, sin transformación de rotación. . . . .	64
D.2.	Resultados de emparejamiento para todos los puntos del test 1. En rojo, con transformación de rotación. En azul, sin transformación de rotación.	65

## ÍNDICE DE FIGURAS

---

D.3. Resultados de emparejamiento para los puntos seleccionados en el test 2. En rojo, con transformación de rotación. En azul, sin transformación de rotación. . . . .	66
D.4. Resultados de emparejamiento de todos los puntos del test 2. En rojo, con transformación de rotación. En azul, sin transformación de rotación. . . . .	67
D.5. Resultados de emparejamiento para los puntos seleccionados en el primer caso del test 3 (disminución de la escala). En rojo, con transformación de rotación. En azul, sin transformación de rotación. . . . .	68
D.6. Resultados de emparejamiento de todos los puntos para el primer caso del test 3 (disminución de la escala). En rojo, con transformación de rotación. En azul, sin transformación de rotación. . . . .	69
D.7. Resultados de emparejamiento para los puntos seleccionados en el caso del test 3 (aumento de escala). En rojo, con transformación de rotación. En azul, sin transformación de rotación. . . . .	70
D.8. Resultados de emparejamiento de todos los puntos para el segundo caso del test 3 (disminución de la escala). En rojo, con transformación de rotación. En azul, sin transformación de rotación. . . . .	71
E.1. Comparison of the theoretical formulas to calculate the ellipse semiaxis in which the sphere is projected (red) with the results of a simulation using a camera model with radial distortion (blue) The prediction for the minor semiaxis (a) is more accurate than the one for the major semiaxis(b). . . . .	80
E.2. Image sequence taken for test 1 (180° rotation). Selected corners for matching are shown in red . . . . .	81
E.3. Matching results of test 1. In red, with oriented patch. In blue, with not-oriented patch . . . . .	82
E.4. Image sequence taken for test 2 (translation). Selected corners for matching are shown in red . . . . .	82
E.5. Matching results of test 2. In red, with oriented patch. In blue, with not-oriented patch . . . . .	83
E.6. Image sequence taken for test 3 (scale change). Selected corners for matching are shown in red . . . . .	84
E.7. Matching results of test 3 (scale decrease). In red, with scaled patch. In blue, with not-scaled patch . . . . .	84
E.8. Matching results of test 3 (scale increase). In red, with scaled patch. In blue, with unescaled patch. . . . .	85
E.9. SLAM trajectory with correlation threshold 0,8 using the warped patch projected on the XY plane (up) and on the YZ plane (down) The red dots are the map features . . . . .	86
E.10. GPS trajectory (red) and SLAM trajectory (green) superposed on the satellite image of the Campus of Bovisa (Milan) where the sequences were acquired . . . . .	86





# Índice de tablas

2.1. Parámetros del modelo de la esfera [4] . . . . .	12
5.1. Emparejamientos y puntos en el mapa final por punto inicialziado . .	38
C.1. Datos de calibración de las cámaras omnidireccionales . . . . .	59
E.1. Matchings and features in map per initialised feature . . . . .	85



# Bibliografía

- [1] Henrik Andreasson, André Treptow, and Tom Duckett. Self-localization in non-stationary environments using omni-directional vision. *Robot. Auton. Syst.*, 55:541–551, July 2007.
- [2] M. Sanjeev Arulampalam, Simon Maskell, and Neil Gordon. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.
- [3] Simon Baker and Shree K. Nayar. A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35:175–196, 1999.
- [4] J. Barreto and H. Araujo. Issues on the geometry of central catadioptric image formation. In *In CVPR*, pages 422–427, 2001.
- [5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110:346–359, June 2008.
- [6] J. S. Chahl and M. V. Srinivasan. A complete panoramic vision system, incorporating imaging, ranging, and three dimensional navigation. In *IEEE Workshop on Omnidirectional Vision (OMNIVIS'00)*, pages 104–111, 2000.
- [7] R. Chang, S.H. Ieng, and R. Benosman. Auto-organized visual perception using distributed camera network. *57(11):1075–1082*, November 2009.
- [8] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel. 1-Point RANSAC for EKF Filtering: application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27(5):609–631, October 2010.
- [9] Javier Civera, Andrew J. Davison, and J. M. M. Montiel. Inverse depth parametrization for monocular slam. *IEEE Transactions on Robotics*, 24(5):932–945, October 2008.
- [10] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *ICCV*, 2003.
- [11] Martin A. Fischler and Robert C. Bolles. Readings in computer vision: issues, problems, principles, and paradigms. chapter Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, pages 726–740. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

- 
- [12] Christopher Geyer and Konstantinos Daniilidis. A unifying theory for central panoramic systems and practical applications. In *ECCV (2)*, pages 445–461, 2000.
- [13] J. J. Guerrero, A. C. Murillo, and C. Sagues. Localization and matching using the planar trifocal tensor with bearing-only data. *IEEE Transactions on Robotics*, 24(2):494–501, 2008.
- [14] Richard Hartley and Andrew Zisserman. *Multiple View geometry in Computer vision*. Cambridge university press, 2000.
- [15] Marko Heikkila, Matti Pietikainen, and Cordelia Schmid. Description of interest regions with local binary patterns. *Pattern Recognition*, 42(3):425–436, mar 2009.
- [16] J. Neira J. A. Castellanos, J. M. M. Montiel and J. D. Tardós. *IEEE Transactions on Robotics and Automation*, 15(5):pp. 948 – 952, October 1999.
- [17] Sing Bing Kang. Catadioptric self-calibration. In *Computer Vision and Pattern Recognition*, pages 1201–1207, 2000.
- [18] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, November 2004.
- [19] Huimin Lu and Zhiqiang Zheng. Two novel real-time local visual features for omnidirectional vision. *Pattern Recogn.*, 43:3938–3949, December 2010.
- [20] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*, April 2007.
- [21] L.M. Paz, P. Jensfelt, J.D. Tardós, and J. Neira. EKF SLAM updates in  $O(n)$  with Divide and Conquer SLAM. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'07)*, Rome, Italy, April 2007.
- [22] A. Rituerto, L. Puig, and J. J. Guerrero. Comparison of omnidirectional and conventional monocular systems for visual slam (best paper award). In *OMNIVIS - 10th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, Zaragoza, Spain, 2010.
- [23] Alejandro Rituerto, Luis Puig, and J. J. Guerrero. Visual slam with an omnidirectional camera. In *Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR '10*, pages 348–351, Washington, DC, USA, 2010. IEEE Computer Society.
- [24] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32:105–119, 2010.
- [25] D. Scaramuzza, F. Fraundorfer, and R. Siegwart. Real-time monocular visual odometry for on- road vehicles with 1-point ransac. In *International Conference on Robotics and Automation (ICRA)*, pages 4293–4299, 2009.

## BIBLIOGRAFÍA

---

- [26] Davide Scaramuzza, Nicolas Criblez, Agostino Martinelli, and Roland Siegwart. Robust Feature Extraction and Matching for Omnidirectional Images. In *6th International Conference on Field and Service Robotics - FSR 2007*, volume 42 of *Springer Tracts in Advanced Robotics*, Chamonix, France, 2007. Springer.
- [27] Davide Scaramuzza and Agostino Martinelli. A toolbox for easily calibrating omnidirectional cameras. In *In Proc. of the IEEE International Conference on Intelligent Systems, IROS06*, pages 5695–5701, 2006.
- [28] G. Scotti, L. Marcenaro, C. Coelho, F. Selvaggi, and C. S. Regazzoni. Dual camera intelligent sensor for high definition 360 degrees surveillance. *Vision, Image and Signal Processing*, 152(2):250–257, 2005.
- [29] Tomás Svoboda and Tomás Pajdla. Matching in catadioptric images with appropriate windows, and outliers removal. In *Proceedings of the 9th International Conference on Computer Analysis of Images and Patterns, CAIP '01*, pages 733–740, London, UK, 2001. Springer-Verlag.
- [30] Tomas Svoboda and Tomas Pajdla. Epipolar geometry for central catadioptric cameras, 2002.
- [31] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.
- [32] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [33] Christian Toepfer and Tobias Ehlgén. A unifying omnidirectional camera model and its applications. In *ICCV*, pages 1–5, 2007.