

## Creación de un modelo visual del entorno basado en imágenes omnidireccionales

A. Gil\*, D. Valiente, O. Reinoso, J.M. Marín

Universidad Miguel Hernández de Elche.  
Avda. de la Universidad s/n, Ed. Quorum V  
03202 Elche (Alicante), España

### Resumen

En este artículo abordamos el problema de la construcción del mapa visual de un entorno mediante un robot móvil, ubicándose, por tanto, en el ámbito del problema de SLAM (*Simultaneous Localization and Mapping*). La solución presentada en este artículo se fundamenta en el uso de un conjunto de imágenes para representar el entorno. De esta manera, la estimación del mapa se plantea como el cálculo de la posición y orientación de un conjunto de vistas omnidireccionales obtenidas del entorno. La idea desarrollada se separa de la concepción habitual de mapa visual, en la que el entorno está representado mediante un conjunto de puntos tridimensionales definidos respecto de un sistema de referencia. En el caso presentado, se considera que el robot está equipado con un único sensor visual omnidireccional que permite detectar un conjunto de puntos de interés de las imágenes que, a continuación, son representados mediante un descriptor visual. El proceso de construcción del mapa se puede resumir de la siguiente manera: cuando el robot se mueve por el entorno captura imágenes omnidireccionales y extrae un conjunto de puntos de interés de cada una de ellas. A continuación, se buscan correspondencias entre la imagen capturada y el resto de imágenes omnidireccionales existentes en el mapa. Si el número de correspondencias encontradas entre la imagen actual y alguna de las imágenes del mapa es suficiente, se calcula una transformación consistente en una rotación y translación. A partir de estas medidas podemos deducir la localización del robot con respecto a las imágenes almacenadas en el mapa. Se presentan resultados obtenidos en un entorno simulado que validan la idea presentada. Además, se han obtenido resultados experimentales utilizando datos reales que demuestran la validez de la solución presentada. Copyright © 2012 CEA. Publicado por Elsevier España, S.L. Todos los derechos reservados.

**Palabras Clave:** SLAM, robótica móvil, visión omnidireccional

### 1. Introducción

La construcción de mapas es uno de los problemas fundamentales en el área de la Robótica Móvil, ya que una gran cantidad de aplicaciones se fundamentan en la existencia de un mapa (Aracil et al., 2008). Para construir el mapa, el robot debe desplazarse por el entorno mientras adquiere información de él. Frecuentemente, la información de la que dispone el robot para realizar el mapa consiste en un conjunto de lecturas de odometría y un conjunto de poses. La naturaleza acumulativa del error existente en la odometría implica un problema de localización, en consecuencia, se genera el problema de construir un mapa mientras, simultáneamente, el robot se localiza dentro de él. Por tanto, el conjunto de algoritmos desarrollados para

esta tarea se agrupan bajo las siglas de SLAM (*Simultaneous Localization and Mapping*).

En la literatura existen un gran número de trabajos que proponen la utilización de cámaras para la construcción de mapas. Estas soluciones se denominan generalmente SLAM visual. A su vez, en este grupo podemos encontrar diversas alternativas que se diferencian en aspectos como el tipo de cámara utilizada, ya sea una única cámara, un par estéreo o una única cámara omnidireccional. Otro factor diferenciador es el tipo de información visual extraída de las imágenes. También se clasifican según extraigan *landmarks visuales* de las imágenes o utilicen un descriptor de apariencia global de las imágenes. En el caso de descriptores visuales, se encuentra un gran número de soluciones basadas en descriptores SIFT (Lowe, 2004) y SURF (Bay et al., 2006) en el ámbito del SLAM visual. La representación utilizada para definir el mapa: en este caso, se encuentran métodos que representan la posición 3D de un conjunto de *landmarks* visuales (Civera et al., 2008; Andrew J. Davison et al., 2004), o bien métodos que estiman un subconjunto de las poses del robot (Paya et al., 2009). El algoritmo de SLAM

\* Autor en correspondencia.

Correos electrónicos: [arturo.gil@umh.es](mailto:arturo.gil@umh.es) (A. Gil),  
[dvaliente@umh.es](mailto:dvaliente@umh.es) (D. Valiente), [o.reinoso@umh.es](mailto:o.reinoso@umh.es) (O. Reinoso),  
[jmarin@umh.es](mailto:jmarin@umh.es) (J.M. Marín)  
URL: [arvc.umh.es](http://arvc.umh.es) (A. Gil)

utilizado: principalmente se encuentran soluciones basadas en el filtro de Kalman, filtros de partículas. Por ejemplo, en (Gil et al., 2006) se utiliza un par estéreo de cámaras calibradas para obtener medidas relativas de distancia a un conjunto de marcas visuales. El mapa está definido por un conjunto de marcas visuales, estando cada una acompañada de un descriptor visual basado en la transformada SIFT (Lowe, 2004). Se emplea un algoritmo basado en un filtro de partículas *Rao-Blackwell* para estimar el mapa y el camino seguido por el robot (Montemerlo et al., 2002). Una solución diferente la encontramos en (Civera et al., 2008), donde se utiliza una única cámara para construir un mapa tridimensional del entorno, constituido por un conjunto de puntos de interés extraídos con el detector de esquinas de Harris (Harris and Stephens, 1988) y descritos por una sub-ventana de niveles de gris. Se capturan imágenes con gran frecuencia mientras la cámara es movida a mano. Cada uno de los puntos 3D detectados se representa mediante un vector adimensional y una escala. La posición 3D de los puntos se estima con bastante precisión en base a un filtro EKF al observar las marcas visuales desde puntos de vista separados por una línea base suficiente. Debido a que una única cámara no nos permite obtener observaciones de la distancia hasta los puntos detectados, la inicialización de la posición tridimensional de las *landmarks* plantea un problema. Este hecho inspiró una parametrización inversa de la profundidad para representar los puntos en el filtro de Kalman (Civera et al., 2008). Según (Andrew J. Davison et al., 2004) los resultados de SLAM visual utilizando una única cámara son mejores cuando se utiliza una óptica con un gran ángulo de visión, hecho que sugiere la utilización de una cámara omnidireccional para la creación del mapa, ya que, en este caso el ángulo de visión horizontal es máximo. Sin embargo, el empleo de cámaras omnidireccionales en aplicaciones de SLAM visual no es demasiado frecuente. Por ejemplo, (Joly and Rives, 2010) estiman con una única cámara omnidireccional y una variación del *Information Filter* estando cada punto modelado mediante una parametrización inversa de la profundidad (Civera et al., 2008). En (Jae-Hean and Myung Jin, 2003) dos cámaras omnidireccionales se combinan para obtener un sensor estéreo con un gran ángulo de visión. Las medidas de distancia obtenidas se integran en un filtro EKF para construir el mapa. En (Scaramuzza et al., 2009) se presenta un método para extraer el movimiento relativo entre dos imágenes omnidireccionales. En este caso, los resultados no se emplean para construir un mapa sino para estimar una odometría visual.

En el caso presentado aquí, consideramos el caso en el que un único robot explora el entorno. El robot está equipado con una única cámara omnidireccional, según se muestra en la Figura 1(a). Cuando el robot se mueve por el entorno, captura imágenes omnidireccionales y extrae un conjunto de puntos de interés de ellas. A continuación, busca correspondencias con el resto de imágenes omnidireccionales existentes en el mapa. Si se encuentra un número suficiente de correspondencias entre las imágenes, se calcula una rotación y translación (salvo un factor de escala) entre ambas imágenes (Scaramuzza et al., 2009). Estas medidas se integrarán en un filtro de Kalman extendido (EKF) para deducir la localización del robot en el mapa, así como la posición del robot cuando capturó cada una de las

imágenes. El cálculo de la rotación y translación se detalla en el apartado 3. En la Figura 1(b) se presentan dos imágenes omnidireccionales donde se han indicado un conjunto de correspondencias. El cálculo de transformación consiste en la obtención de los ángulos  $(\phi, \beta)$  indicados a partir de las correspondencias de puntos entre ambas imágenes, quedando el factor de escala en la transformación  $\rho$  indeterminado. El proceso de cálculo y la integración de las medidas obtenidas entre imágenes se presenta en el apartado 2.

En la mayoría de los trabajos de SLAM visual que se encuentran en la literatura, el mapa se representa mediante un conjunto de puntos tridimensionales que representan elementos del entorno (Gil et al., 2006, 2010; Civera et al., 2008; Davison and Murray, 2002; Ballesta et al., 2010). Típicamente, estos puntos son obtenidos mediante un algoritmo de detección de puntos de interés como, por ejemplo, Harris (Civera et al., 2008) y suelen acompañarse de un descriptor visual más o menos invariante de la apariencia visual del punto. Al conjunto del punto y del descriptor se le denomina *visual landmark* en la literatura anglosajona y se traduce al castellano como marca visual. Independientemente del algoritmo de SLAM utilizado, en los trabajos mencionados el proceso de cálculo del mapa implica la estimación de la posición de cada una de las marcas del mapa. En contraposición con este tipo de mapa, en este artículo exponemos una concepción del mapa diferente. El mapa está formado por la posición y orientación de un conjunto de vistas del entorno. Cada vista se define como la posición y orientación de la cámara cuando esta capturó la imagen en el entorno, junto con un conjunto de puntos de interés y descriptores visuales. El proceso de cálculo plantea la estimación de la posición y orientación de todas las vistas del mapa. La construcción del mapa se resume a continuación: supóngase que el robot parte desde el origen del sistema global de referencia. En ese instante, captura una *vista* inicial. Mientras el robot se mueve en las cercanías de esta vista inicial captura imágenes y encuentra puntos correspondientes entre la imagen actual y la vista inicial, calculando una rotación y translación y localizándose respecto de la vista inicial. Cuando el robot se aleja de la vista inicial, no será capaz de encontrar puntos correspondientes. En este momento iniciará una nueva vista en el mapa. Esta nueva vista permitirá la localización del robot en su cercanía.

La solución presentada en este artículo presenta algunas ventajas si la comparamos con otras soluciones de SLAM visual previas. La ventaja principal radica en la compacidad de la representación del entorno. Soluciones como (Andrew J. Davison et al., 2004; Civera et al., 2008) estiman la posición de las *landmarks* visuales, así como la posición y orientación de la cámara, utilizando 6 variables para tal fin, con lo que el vector de estado del problema de SLAM crece rápidamente con el número de *landmarks* almacenadas en el mapa. Este hecho plantea un problema para la mayoría de algoritmos de SLAM, haciendo que los tiempos de cálculo aumenten de forma cuadrática con el número de *landmarks* en el mapa. En la solución presentada en este artículo, únicamente se estima la posición de un reducido conjunto de vistas. Cada vista encapsula información de un área del entorno en forma de un conjunto de puntos de interés. Según se demostrará mediante experimentos reales y en

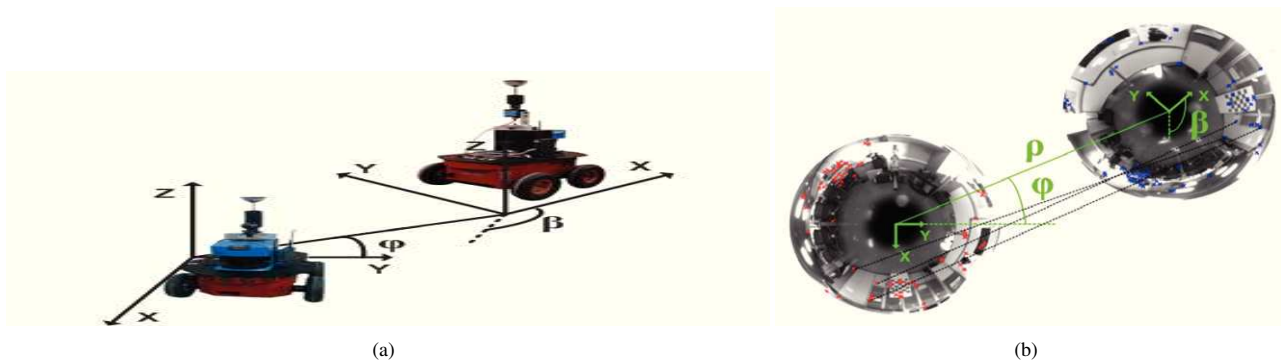


Figura 1: La Figura 1(a) muestra la configuración del sensor usado en los experimentos. La Figura 1(b) presenta dos imágenes omnidireccionales reales, con varias correspondencias indicadas.

simulación, esta representación del mapa es más eficiente y requiere de un menor coste computacional para su cálculo, aún así permite una localización precisa del robot.

A pesar de que el coste computacional sea un factor restrictivo, en el apartado 3 detallamos un algoritmo que puede ser utilizado para calcular la observación con una frecuencia alta y permite la realización de SLAM en tiempo real. En este caso, el cálculo de la transformación entre dos imágenes depende sólo del número de correspondencias encontradas en las imágenes y no del número de vistas existentes en el mapa, con lo que es un tiempo constante en cada iteración del filtro.

Durante los experimentos se han utilizado las características SURF para la detección y descripción de los puntos. La solución presentada no está limitada al uso de este detector y descriptor visual. El fundamento del uso de este descriptor se basa en un estudio anterior sobre detectores y descriptores visuales en su aplicación al SLAM visual (Gil et al., 2010; Ballesta et al., 2010), donde las características SURF presentaron muy buenas cualidades para esta tarea.

Se presentarán un conjunto de resultados obtenidos en simulación y utilizando datos reales que permiten demostrar la validez de la solución de SLAM visual presentada.

El resto del artículo se organiza de la siguiente manera. Primero, el apartado 2 se describe el proceso de SLAM. A continuación, se describe el algoritmo usado para estimar la transformación entre imágenes omnidireccionales en el apartado 3. Seguidamente se aborda el problema de la asociación de datos en el apartado 4. El apartado 5 presenta los principales resultados experimentales. Finalmente, las conclusiones más relevantes se exponen en el apartado 6.

## 2. Construcción de mapas (SLAM)

En el método de SLAM (*Simultaneous Localization and Mapping*, o construcción de mapas y localización simultánea) que se presenta aquí, cada imagen omnidireccional integrada en el mapa se denominará *vista*, para diferenciarla del concepto de *landmark* visual utilizado comúnmente en este ámbito. Es importante recalcar que una *landmark* visual corresponde a un punto físico en el entorno, como, por ejemplo, una esquina sobre una pared. Sin embargo, una *vista* representa la información

visual obtenida desde una pose en particular del entorno. En consecuencia representamos una *vista* mediante una pose donde se capturó la imagen en el entorno, acompañada de la posición bi-dimensional de los puntos detectados en dicha imagen junto con sus descriptores visuales. El mapa estará integrado por un número finito de vistas capturadas por el robot desde poses diferentes. Cuanto mayor es el número de vistas utilizadas, más completa será la representación del mapa, pero mayor el número de variables a estimar. Según se demostrará en la parte experimental, un conjunto reducido de vistas permite modelar la mayoría de entornos, ya que cada vista permite al robot localizarse en un área cercana. En nuestro caso, según se indicará en el apartado 2.3, se incluyen nuevas vistas cuando la apariencia global de la imagen actual capturada por el robot difiere en gran medida de la apariencia de cualquiera de las vistas existentes en el mapa.

Consideramos que esta representación del entorno se puede emplear para la estimación de un mapa mediante algoritmos de SLAM diferentes, bien métodos *online* como EKF, FastSLAM o bien *offline*, como, por ejemplo, *Stochastic Gradient Descent* (Grisetti et al., 2007). En este artículo presentamos como ejemplo la estimación del mapa mediante un filtro EKF y probamos que se pueden obtener resultados correctos con datos reales.

Igualmente, esta representación del mapa y el modelo de observación pueden ser utilizados para la creación de un mapa basado en vistas capturadas mediante una única cámara estándar. La razón fundamental que justifica el empleo de una cámara omnidireccional es la habilidad de adquirir una visión global del entorno con una única imagen.

### 2.1. Representación y Estimación del Mapa mediante EKF

A continuación definimos con precisión la representación utilizada para la estimación del mapa del entorno mediante un filtro EKF. La pose del vehículo en el instante  $t$  se indicará como  $x_v = (x_v, y_v, \theta_v)^T$ . Cada vista  $i$  está representada por su pose  $x_{i_i} = (x_{i_i}, y_{i_i}, \theta_{i_i})^T$ , su incertidumbre  $P_{i_i}$  y un conjunto de  $M$  puntos de interés  $p_j$  expresados en coordenadas de imagen. Cada punto de interés está asociado con un descriptor visual  $d_j$ ,  $j = 1, \dots, M$ . En total, consideramos que en el instante  $t$  existen  $N$  vistas  $x_{i_i}$  incluidas en el mapa, por tanto  $i = 1, \dots, N$ .

En la Figura 2 se ilustra este tipo de mapa, donde se indica la posición de un conjunto de vistas. Por ejemplo, la vista *A* se capturó desde la pose particular  $x_{l_A} = (x_{l_A}, y_{l_A}, \theta_{l_A})^T$  en el mapa y tiene un conjunto de *M* puntos de interés asociados. En el caso presentado, la vista *A* permite la localización del robot en sus inmediaciones. La vista *B* se utiliza para modelar una de las estancias y permitirá la localización del robot en sus cercanías. Finalmente, las vistas *C*, *D* y *E* modelan el resto de estancias del entorno. Lógicamente, se deberá establecer un método para la inicialización de las vistas cuando el robot no sea capaz de establecer correspondencias con ninguna de las vistas existentes en el mapa, o bien cuando las vistas existentes no le permitan localizarse con exactitud. En el apartado 2.3 se presenta un método sencillo para realizar esta tarea.

Para la estimación del mapa y de la posición del vehículo en el instante *t*, definimos un vector de estado para el filtro EKF como:

$$\bar{x}(t) = [x_v, x_{l_1}, x_{l_2}, \dots, x_{l_N}]^T \quad (1)$$

donde *N* es el número de vistas que existen en el mapa,  $x_v$  la pose del robot y  $x_{l_i}$  la pose de la vista *i*.

La relación entre el estado en el instante *t* + 1 y el estado actual es la siguiente:

$$\bar{x}(t+1) = F(t)\bar{x}(t) + u(t+1) + v(t+1) \quad (2)$$

donde *F*(*t*) contiene la información relativa a la transición entre estados, *u*(*t* + 1) es el vector de control del movimiento que genera la odometría del robot y *v*(*t* + 1) es el ruido que se añade al sistema, el cual es de tipo gaussiano y con correlación nula.

Del mismo modo puede definirse una relación lineal entre la observación realizada por el sistema sensorial en un instante *t* de una vista *i*,  $z_i(t)$ , con la variable de estado.

$$z_i(t) = H_i(t)\bar{x}(t) + w_i(t) \quad (3)$$

donde  $H_i(t)$  representa la relación entre  $\bar{x}(t)$  y  $z_i(t)$ , y  $w_i(t)$  es el ruido aleatorio que se genera en el proceso, el cual es gaussiano y de covarianza *R*(*t*).

A continuación hay que diferenciar las tres etapas fundamentales del procedimiento de filtrado. En primer lugar se lleva a cabo una predicción del estado a estimar  $\hat{x}(t)$ , y en base a ésta se obtiene la predicción de la observación  $\hat{z}_i(t)$ :

$$\hat{x}(t+1|t) = F(t)\hat{x}(t|t) + u(t) \quad (4)$$

$$\hat{z}_i(t+1|t) = H_i(t)\hat{x}(t+1|t) \quad (5)$$

$$P(t+1|t) = F(t)P(t|t)F^T(t) + Q(t) \quad (6)$$

donde *P*(*t*|*t*) y *P*(*t* + 1|*t*) son matrices de covarianza que representan la incertidumbre de la estimación en *t* y *t* + 1 respectivamente.

En la segunda etapa se realiza la observación  $z_i(t)$  de una determinada vista *i* del mapa, cuya asociación de datos se asume correcta, y mediante la cual se puede definir el concepto de innovación, como la variación entre la estimación a priori y la medida de observación:

$$v_i(t+1) = z_i(t+1) - \hat{z}_i(t+1|t) \quad (7)$$

$$S_i(t+1) = H_i(t)P(t+1|t)H_i^T(t) + R_i(t+1) \quad (8)$$

donde  $S_i(t+1)$  representa la covarianza de la innovación.

Finalmente, en la tercera etapa se actualiza la estimación obtenida en la primera etapa según el valor de la innovación obtenida en la segunda etapa, obteniendo así la solución al filtro para el instante *t* + 1:

$$\hat{x}(t+1|t+1) = \hat{x}(t+1|t) + K_i(t+1)v_i(t+1) \quad (9)$$

$$P(t+1|t+1) = P(t+1|t) - K_i(t+1)S_i(t+1)K_i^T(t+1) \quad (10)$$

donde en este caso  $K_i(t+1)$  se corresponde con la ganancia del filtro EKF, obteniéndose del siguiente modo:

$$K_i(t+1) = P(t+1|t)H_i^T(t)S_i^{-1}(t+1) \quad (11)$$

Para el caso que nos ocupa, inicializamos las matrices de covarianza de ruido *Q*(*t*) y *R*(*t*) que introducen la odometría y el modelo de observación respectivamente. La primera de ellas se establece en base a los parámetros de ruido conocidos que genera la odometría del robot, y la segunda se determina en base a medidas experimentales, tal y como se detalla en el apartado 5. La odometría *u*(*t*), se emplea para la obtención de la predicción, conjuntamente con el estado anterior, tal y como se deduce de la ecuación 4. La matriz de incertidumbre del mapa estimado, *P*(*t*), tiene en cuenta el ruido de la odometría según la ecuación 6, y el ruido introducido por el sensor visual a la hora de realizar una medida de observación, como puede comprobarse en las ecuaciones 8 y 10. En particular, el modelo propuesto de observación  $z_i(t)$  se detalla a continuación.

## 2.2. Modelo de Observación

El modelo de observación nos permite obtener una información relativa para la estimación indirecta de la pose del robot y de las vistas. En lo siguiente se asume que el robot se encuentra en una posición en el entorno y captura una imagen omnidireccional  $I_i$ . A continuación, suponemos que hemos sido capaces de encontrar un conjunto de puntos correspondientes entre  $I_i$  y una de las vistas almacenadas en el mapa  $I_{l_i}$ . Según se describirá en el apartado 3, obtenemos una observación  $z_i(t)$ :

$$z_i(t) = \begin{pmatrix} \phi \\ \beta \end{pmatrix} = \begin{pmatrix} \text{atan}(\frac{y_{l_i} - y_v}{x_{l_i} - x_v}) - \theta_v \\ \theta_{l_i} - \theta_v \end{pmatrix} \quad (12)$$

donde el ángulo  $\phi$  es la orientación con la que la vista *i* es observada desde el sistema de referencia móvil asociado al robot y  $\beta$  es la orientación relativa entre ambas imágenes. La vista *i* está representada por  $x_{l_i} = (x_{l_i}, y_{l_i}, \theta_{l_i})$ , mientras que la pose del robot está descrita por  $x_v = (x_v, y_v, \theta_v)$ . Ambas medidas ( $\phi, \beta$ ) se presentan en la Figura 1(a).

## 2.3. Inicialización de Nuevas Vistas

Según se dijo, es necesario proporcionar un método para incluir nuevas vistas en el mapa cuando estas sean necesarias. En nuestro caso, se incluye una nueva vista en el mapa cuando la apariencia de la imagen actual es muy diferente de cualquiera

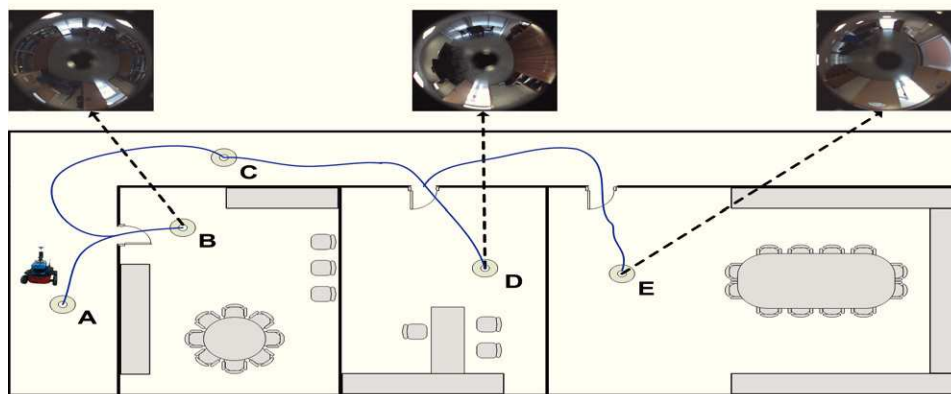


Figura 2: La figura presenta la idea básica para la construcción del mapa. El robot comienza la exploración en el punto A y almacena una vista  $I_A$  en el origen. A continuación se mueve. Cuando no se encuentran correspondencias entre la imagen actual e  $I_A$ , una nueva vista es creada en la posición actual del robot, B. El proceso continúa hasta que el entorno queda completamente representado.

de las vistas almacenadas en el mapa. El cálculo de la apariencia global se aproxima mediante la siguiente ratio:

$$R = \frac{2K}{n_A + n_B} \quad (13)$$

que calcula el grado de similitud entre las vistas A y B, siendo K el número total de correspondencias puntuales entre A y B, mientras que  $n_A$  y  $n_B$  son el número de puntos detectados en las imágenes A y B respectivamente.

El robot decidirá incluir una nueva vista en el mapa cuando la ratio R cae por debajo de un valor predefinido. Así pues, la inicialización de las vistas depende de un único factor de similitud R. Si se selecciona un valor de R alto, se incluirá un gran número de nuevas vistas en el mapa, aumentando la precisión con la que podremos localizar al robot pero incrementando el coste computacional necesario para calcular el mapa. En el caso opuesto, si se selecciona un valor de R bajo, el número de vistas será menor, reduciéndose el tiempo de cómputo pero también reduciendo la precisión con la que podemos localizar al robot en el mapa.

En la inicialización de cada vista, la pose  $x_i$  y la incertidumbre asociada se obtienen de la estimación de  $x_v$  en el instante actual  $t$  y de su submatriz de covarianza asociada, ya que en el instante  $t$  la posición de la vista y la posición del robot coinciden.

### 3. Cómputo de la Transformación entre Imágenes Omnidireccionales

En este apartado proponemos un método para calcular la transformación entre dos imágenes omnidireccionales. La transformación se puede calcular salvo un factor de escala y está representada mediante los ángulos  $(\beta, \phi)$ , según se indicó en el apartado 2.2. Estos ángulos representan la posición relativa del robot a una de las vistas del mapa y permiten su localización. Para su obtención deben detectarse puntos característicos en ambas imágenes y encontrar sus correspondencias aplicando

la condición de epipolaridad. Los esquemas tradicionales, tales como (Kawanishi et al., 2008; Nister, 2003; Stewenius et al., 2006) resuelven el caso general con 6 GDL, mientras que en nuestro caso, asumiendo que el movimiento del robot se restringe a un plano, podemos limitar el cálculo a 4 variables de la matriz esencial, reduciendo de este modo el coste computacional.

#### 3.1. Detección de Puntos Significativos y Correspondencias

Durante las pruebas experimentales se han empleado las características SURF (Bay et al., 2006) con el fin de obtener puntos de interés y correspondencias entre imágenes. Según el estudio presentado en (Gil et al., 2010; Ballesta et al., 2010), el detector y descriptor SURF obtuvo excelentes resultados en términos de robustez de los puntos detectados y de invarianza del descriptor al compararse con otros métodos empleados en el ámbito de SLAM visual. La extracción de puntos de interés y su descripción se realizan a partir de una imagen panorámica, si bien una vez obtenidos los puntos, se trabaja con sus coordenadas en la esfera unidad sobre el sistema de referencia original. Para ello, como primer paso, se transforma la imagen omnidireccional capturada con la cámara a una vista panorámica y se extraen un conjunto de puntos característicos. A continuación, para cada uno de estos puntos se calcula un descriptor SURF. Según se comprueba experimentalmente, ante un movimiento de la cámara, la variación en la apariencia local de los puntos (y, por tanto, la variación en el descriptor) es menor en la imagen panorámica que en la omnidireccional. En la Figura 3.1 se presenta un ejemplo de imagen omnidireccional capturada por el sistema catadióptrico y su transformación a imagen panorámica. De esta manera se consigue aumentar el número de correspondencias válidas entre imágenes. Hay que destacar, que finalmente los puntos detectados en la imagen panorámica se re proyectan sobre la esfera unidad en las coordenadas de la vista original, es decir sobre la vista omnidireccional, y se almacenan junto con los descriptores calculados. El cambio a vista

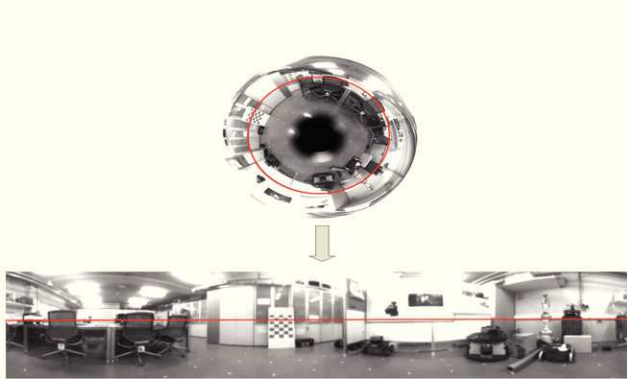


Figura 3: La figura muestra una imagen omnidireccional y su vista panorámica correspondiente. En la figura, la línea dibujada representa la posición de los mismos puntos en ambas imágenes. Una vez detectados puntos de interés en la vista panorámica, se realiza la reconversión de estos puntos sobre la esfera unidad en el sistema de referencia original.

panorámica se realiza únicamente con la intención de incrementar el número de puntos detectados, puesto que este modelo ha sido planteado para obtener transformaciones entre imágenes omnidireccionales.

### 3.2. Cómputo de la Transformación

Una vez detectados los puntos SURF en cada una de las vistas y suponiendo conocidas un conjunto de correspondencias entre imágenes, ha de establecerse un proceso para calcular los ángulos relativos  $\beta$  y  $\phi$ .

#### 3.2.1. Geometría Epipolar

La condición de epipolaridad establece la relación entre dos puntos 3D observados desde diferentes vistas. Se puede expresar como:

$$\rho p'^T E p = 0 \quad (14)$$

donde la matriz  $E$  recibe el nombre de matriz esencial. El mismo punto detectado en dos imágenes se expresa como  $p = [x, y, z]^T$  en el sistema de referencia fijo de la primera cámara y  $p' = [x', y', z']^T$  en el de la segunda (considerado móvil). La matriz esencial  $E$  representa una rotación  $R$  y una traslación  $T$  (salvo un factor de escala  $\rho$ ) entre los sistemas de referencia de dos imágenes, con  $E = R \cdot T_x$ . Por tanto los ángulos deseados ( $\beta, \phi$ ), pueden ser obtenidos a partir de los elementos de  $E$ . Debe señalarse que la Geometría Epipolar puede ser usada en imágenes omnidireccionales ya que re proyectamos el sistema 2D del plano imagen a 3D mediante el modelado del espejo hiperbólico de la cámara, a partir de una calibración previa (Scaramuzza et al., 2006). A causa de la ambigüedad en la profundidad, denotamos  $\vec{p}$  and  $\vec{p}'$  en 3D, como los vectores unitarios que indican la dirección de los puntos en los dos sistemas de referencia, ya que la posición 3D no puede ser totalmente definida con una única vista de la escena. De otra manera: el método presentado permite calcular la matriz  $E$  salvo un factor de escala  $\rho$ , el cual a efectos prácticos se elige de manera arbitraria para la resolución del problema. Aun así, en los experimentos con datos reales que se presentan, la escala real del

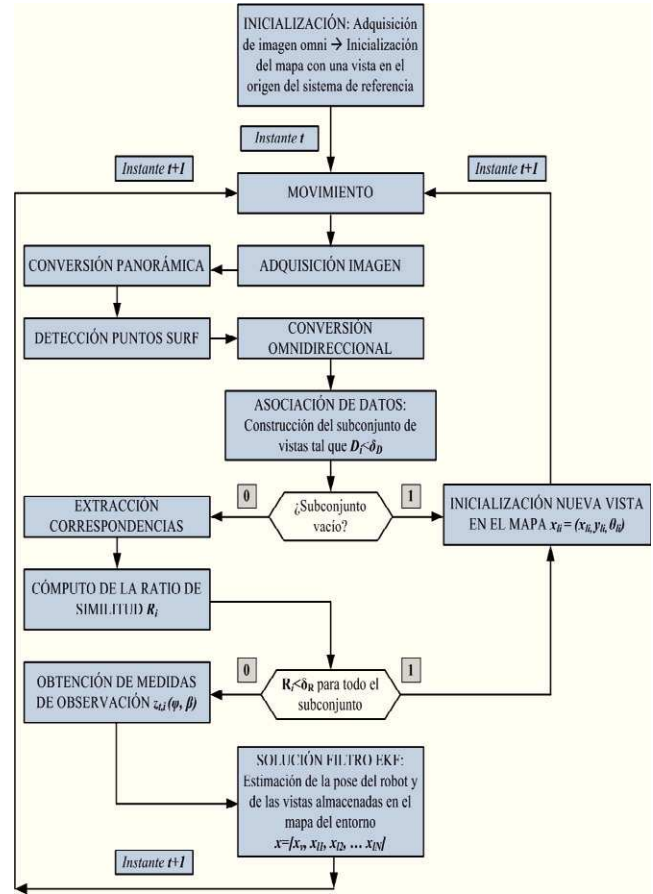


Figura 4: Diagrama de bloques representativo del modelo de SLAM propuesto.

mapa puede determinarse con bastante precisión a partir de las medidas de odometría del robot, resolviendo de este modo la indeterminación del factor de escala  $\rho$ .

Con el fin de obtener  $\beta$  y  $\phi$ , hemos considerado (Hartley and Zisserman, 2004), donde se sugiere el empleo de la matriz de proyección  $P$ , la cual también define la transformación entre imágenes. Se ha adoptado este método por su simplicidad a la hora de calcular las cuatro posibles soluciones del problema. Al estimar una rotación y una traslación sobre un movimiento en el plano XY, sólo son necesarias  $K = 4$  correspondencias para resolver el problema, ya que la matriz  $E$  tiene la siguiente forma:

$$E = \begin{bmatrix} 0 & 0 & e_{13} \\ 0 & 0 & e_{23} \\ e_{31} & e_{32} & 0 \end{bmatrix} \quad (15)$$

En cambio, calculamos  $E$  con un mayor número de puntos a fin de obtener soluciones fiables en presencia de ruido y falsas correspondencias. Además, empleamos un algoritmo RANSAC (Nistér, 2005) para filtrar posibles correspondencias erróneas.

Para el cálculo de  $E$ , primero aplicamos la condición de epipolaridad  $\vec{p}'^T \cdot E \cdot \vec{p} = 0$  sobre  $K$  puntos, y resolvemos la ecuación resultante  $D \cdot \vec{e} = 0$ . Donde,  $D$  es una matriz de coeficientes que se obtiene como resultado de aplicar la ecuación (14) a  $K$

puntos, y  $\vec{e} = [e_{13} \ e_{23} \ e_{31} \ e_{32}]$ . A continuación descomponemos  $E$  mediante SVD:

$$[U|S|V] = SVD(E), \quad (16)$$

que permite calcular:

$$R_1 = [UV^T W], \ R_2 = [UV^T W^T], \ T = [UZU^T] \quad (17)$$

siendo  $W$  y  $Z$  matrices auxiliares (Hartley and Zisserman, 2004) y las posibles rotaciones ( $R_1, R_2$ ) y traslaciones ( $T_{1x}, -T_{1x}$ ). Para obtener las cuatro posibles P-matrices, computamos:

$$P_1 = [R_1|T_{1x}], \ P_2 = [R_1|-T_{1x}], \quad (18)$$

$$P_3 = [R_2|T_{1x}], \ P_4 = [R_2|-T_{1x}], \quad (19)$$

En nuestro caso, las matrices de proyección tienen la forma:

$$P_i = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 & \rho \cos(\phi) \\ \sin(\beta) & \cos(\beta) & 0 & \rho \sin(\phi) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (20)$$

Nótese que  $\beta$ ,  $\phi$  y  $\rho$  pueden tomar diferentes valores que cumplan la condición de epipolaridad (14) debido a la indeterminación del factor de escala  $\rho$ . Esto plantea un proceso de selección de una de las cuatro posibles soluciones descritas en (18) como la correcta. En nuestro caso, hemos utilizado una solución basada en mínimos cuadrados, según se detalla en (Bunschoten and Krose, 2003). Dicho proceso se detalla a continuación:

### 3.2.2. Selección de la Solución

La selección de la solución correcta debe llevarse a cabo mediante un procedimiento inverso. Multiplicamos  $\vec{p}'$  por la inversa de cada una de las cuatro matrices de proyección posibles  $P_i$ , obteniendo así cuatro estimaciones de  $\vec{p}$ . Se asume como solución correcta aquella que genera la estimación con menor desviación respecto de  $\vec{p}$ . Por último,  $\beta$  y  $\phi$  son directamente recuperados a partir de los elementos de  $P$  definidos en (20).

## 4. Asociación de datos

El problema de la asociación de datos reviste gran importancia en el caso general de SLAM visual basado en *landmarks*. Dicho problema se puede enunciar de la siguiente manera: dado un conjunto de observaciones  $z_i(t) = \{z_{i,1}, \dots, z_{i,B}\}$  obtenidas en el instante  $t$ , se deberá decidir cuáles de las *landmarks* del mapa generaron dichas observaciones. Así pues, el resultado del proceso de asociación de datos es un vector de índices  $H = \{j_1, \dots, j_B\}$  donde cada uno de los índices  $j_i \in [1, N+1]$  denota una de las *landmarks* del mapa, siendo  $N$  el número total de *landmarks* en el mapa. Si la observación  $z_i(t)$  no está asociada a ninguna de las *landmarks* del mapa, se inicializará una nueva con índice  $N+1$ . Este proceso de asociación de datos es crucial en SLAM. El caso del SLAM visual basado en *landmarks* visuales es particularmente complejo, ya que en el mapa pueden existir un gran número de marcas visuales y la apariencia de los puntos correspondientes puede variar considerablemente. Por ejemplo, en (Gil et al., 2006) se utiliza una distancia

de Mahalanobis para encontrar un conjunto de candidatos entre las *landmarks* del mapa. A continuación, se elige la correspondencia en función de la similitud entre los descriptores visuales. Otras soluciones más elaboradas para hallar la asociación de datos, como la presentada en (Neira and Tardós, 2001) son de difícil aplicación, debido al coste computacional que implican.

En el caso presentado aquí, basado en el uso de *vistas* omnidireccionales para la construcción del mapa, la asociación de datos se puede abordar de una manera diferente. Consideremos que en un instante  $t$  el robot captura una imagen omnidireccional  $I_i$ . Asumamos que, en ese instante  $t$  existen  $N$  vistas en el mapa  $I_1, I_2, \dots, I_N$ . Primero, se seleccionan un conjunto de vistas cercanas a la vista actual, según las poses de las mismas. Esta selección se realiza en base a la distancia Euclídea:

$$D_i = \sqrt{(x_v - x_{I_i})^T \cdot (x_v - x_{I_i})} \quad (21)$$

La vista  $i$  se incluye en el conjunto de candidatos si  $D_i < \delta_D$ , donde  $\delta_D$  es una distancia elegida experimentalmente. Valores altos de  $\delta_D$  precisan un mayor número de candidatos para realizar la búsqueda, con lo que se incrementa el coste computacional.

A continuación, se busca un conjunto de puntos correspondientes entre la imagen  $I_i$  y cada uno de las imágenes en el grupo de candidatos  $\{I_1, I_2, \dots, I_J\}$ . La búsqueda de puntos correspondientes se realiza teniendo en cuenta la restricción epipolar (14). Finalmente, en base al número de correspondencias encontradas, se calcula la ratio  $R$  (13) y se obtiene una observación  $z_i(t) = (\phi, \beta)$  entre la vista  $I_i$  y la vista  $I_k$  ( $k \in [1, J]$ ) si la ratio  $R$  supera un determinado valor fijado experimentalmente.

De esta manera, cuando la ratio  $R$  es alta, existen un gran número de correspondencias correctas entre la vista actual  $I_i$  y la vista candidato, con lo que la observación  $z_i(t) = (\phi, \beta)$  será precisa. Si la ratio  $R$  es baja, el número de correspondencias entre las imágenes es reducido, con lo que la observación  $z_i(t)$  con gran probabilidad, será incorrecta. De esta manera, en base al factor  $R$  podemos decidir la asociación de datos y la inicialización de nuevas vistas.

## 5. Resultados

Los resultados experimentales se agrupan en dos apartados diferentes. Primero, en el apartado 5.1 presentamos los resultados obtenidos en simulación que permiten validar el esquema de SLAM aquí propuesto. Seguidamente, en el apartado 5.2 mostramos resultados experimentales reales.

### 5.1. SLAM: Resultados en Simulación

Hemos realizado una serie de experimentos en simulación que permiten validar el concepto general de vista y su estimación mediante un algoritmo de SLAM basado en el filtro EKF. Nótese la importancia de asegurar la convergencia de un algoritmo de SLAM basado en EKF, con el modelo de observación presentado en la ecuación 12, ya que el modelo de observación se debe linealizar para incluirlo en el filtro de Kalman. Los experimentos en simulación se han realizado en dos escenarios

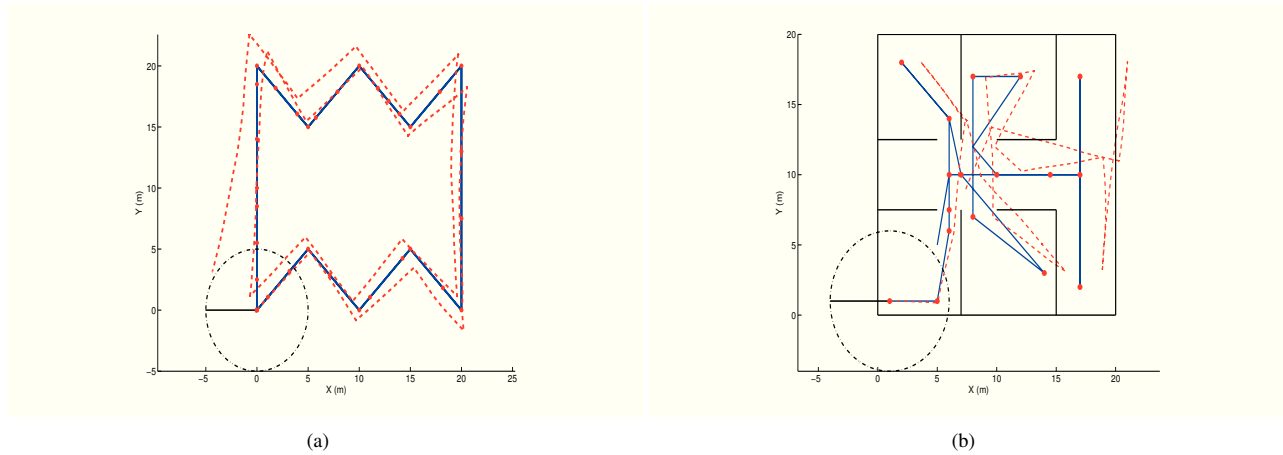


Figura 5: La Figura 5(a) representa el escenario simulado 1. La localización de las distintas vistas en el mapa se representa con puntos. La Figura 5(b) representa el escenario simulado 2.

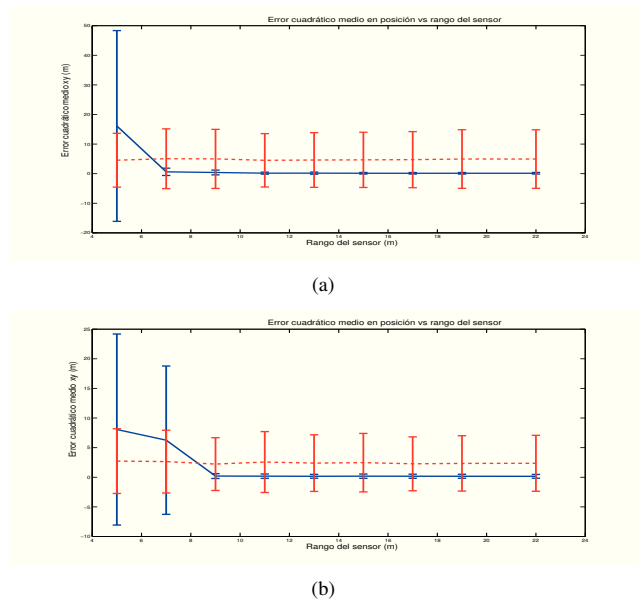


Figura 6: La Figura 6(a) presenta los resultados obtenidos en el escenario simulado 1. La Figura 6(b) presenta los resultados obtenidos en el escenario simulado 2.

virtuales con características diferentes. La Figura 5(a) muestra el escenario simulado 1, donde se simula un entorno en el que no existen obstáculos que impidan la visibilidad sobre las diferentes vistas del mapa. Por otra parte, la Figura 5(b) representa el escenario simulado 2, el cual emula un entorno típico de interior donde existen obstáculos, tales como paredes, que limitan la visibilidad sobre las vistas del mapa. Se asume que el robot puede calcular una observación  $z_i(t)$  con alguna de las vistas del mapa cuando podemos trazar una línea recta entre la pose del robot y la posición de la vista, siempre que estén dentro del radio de observación  $\delta_D$ . En las Figuras, 5(a) y 5(b) se presenta además con línea continua el camino real seguido por el robot, mientras que con línea de puntos se muestran las lecturas de

odometría. Un conjunto de vistas han sido aleatoriamente dispuestas a lo largo de las trayectorias y se muestran con puntos. Nótese que, según se indicó en el apartado 2.3, el emplazamiento de las vistas depende de la similitud entre las imágenes y de la ratio  $R$  elegida. En los dos escenarios, la simulación genera una variación aleatoria de  $R$ , por tanto se está simulando el procedimiento de inicialización de vistas en el mapa por parte del robot a medida que va descubriendo el entorno. Esta simulación de la disposición de vistas emula el comportamiento normal de los experimentos reales, ya que la variación de  $R$  se ha escogido para tal efecto. Puesto que la intención principal de estos primeros experimentos es la validación de la convergencia del algoritmo de SLAM, las imágenes asociadas a las vistas son omitidas. De este modo las observaciones  $z_i(t)$  que realiza el robot también son simuladas con una covarianza obtenida experimentalmente de  $\sigma_\phi = 0, 1 = \sigma_\beta = 0, 1 rad$ . El radio de observación del robot  $\delta_D$  se representa mediante un círculo discontinuo centrado en la pose real del robot.

A continuación presentamos los resultados obtenidos en simulación con el escenario simulado 1. En ambos casos, el robot comienza el proceso de SLAM en el origen y realiza dos vueltas a lo largo de la trayectoria indicada. Las observaciones obtenidas por el robot han sido simuladas según el modelo presentado en la ecuación 12 con un ruido gaussiano simulado mediante una matriz de covarianza  $R(t) = \text{diag}(\sigma_\phi^2 = 0, 1^2 rad^2, \sigma_\beta^2 = 0, 1^2 rad^2)$ . Hemos llevado a cabo una serie de experimentos donde se varía el radio de observación del robot  $\delta_D$ . Los resultados se presentan en la Figura 6(a) y 6(b), donde se muestra el error RMS en la trayectoria frente al radio de observación. Dicho error, representa la desviación cuadrática media tanto de la estimación según el filtro EKF (línea continua), como de la odometría (línea a trazos) comparada con el camino real. El experimento se ha repetido 50 veces, generando aleatoriamente 50 series diferentes de odometría. En las Figuras 6(a) y 6(b) se observa el error RMS medio de los distintos experimentos, así como intervalos de  $2\sigma$ . Según se puede observar en la Figura 6(a), cuando el radio de observación está por debajo de 6

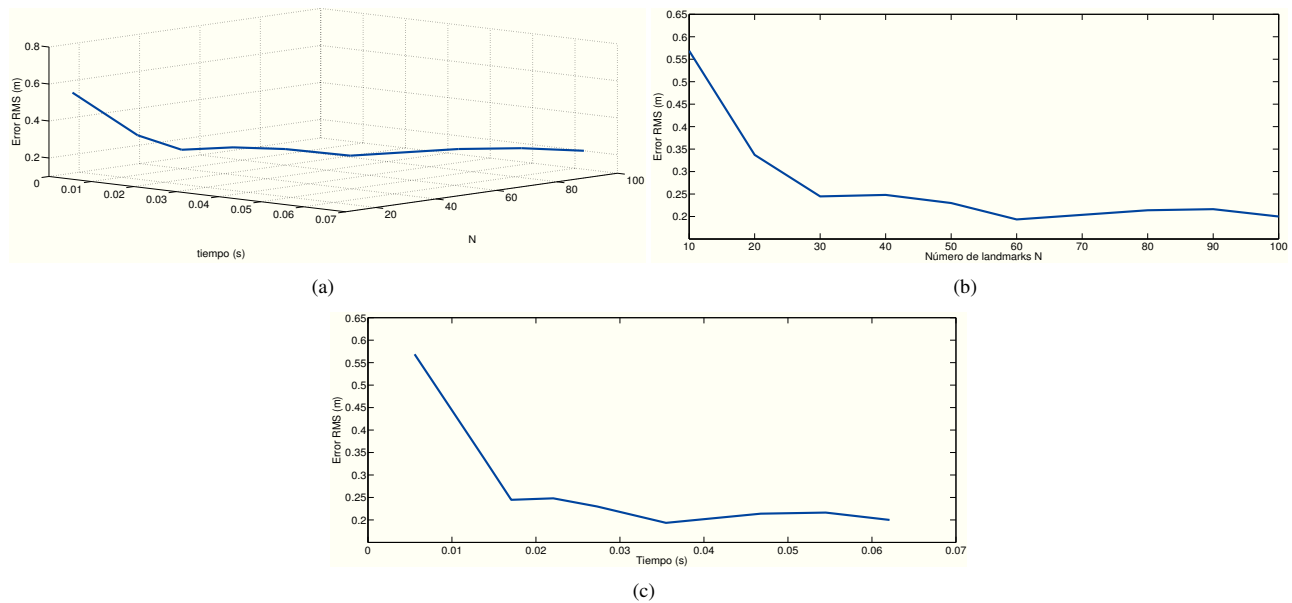


Figura 7: La Figura 7(a) representa el error RMS cometido en el camino frente al número de *landmarks*  $N$  y el tiempo de cómputo medio por iteración del algoritmo de SLAM. La Figura 7(b) representa el error RMS frente al número de *landmarks*  $N$  mientras que la Figura 7(c) muestra el error RMS frente al tiempo de cómputo medio por cada iteración del algoritmo de SLAM.

$m$  la incertidumbre en la posición es alta, denotando que el filtro no ha sido capaz de filtrar el ruido en la odometría. Esto es debido a que la vistas están emplazadas a distancias mayores, y el robot no ha sido capaz de obtener suficientes observaciones. Para valores de radio superiores a  $6\text{ m}$  el error RMS disminuye demostrando la convergencia del filtro. Un resultado similar se presenta en la Figura 6(b), la cual se corresponde con el escenario simulado 2. En este caso se obtienen resultados satisfactorios con valores de  $\delta_D$  por encima de  $9\text{ m}$ . La diferencia existente entre ambos resultados se puede explicar debido a la existencia de elementos en el entorno que limitan la visibilidad entre las vistas, dificultando así la obtención de observaciones.

Es necesario recalcar que los resultados obtenidos dependen fuertemente del emplazamiento y número de vistas. Si se sitúan más vistas en el entorno se consigue un cálculo más preciso tanto del mapa como de la trayectoria, a cambio de un mayor coste computacional. Con esta idea en mente se realizaron un conjunto de simulaciones en las que se mantuvo constante el radio de observación  $\delta_D$  y se varió el número  $N$  de vistas en el mapa mientras se media el tiempo necesario para realizar el experimento. Cada simulación en el escenario 1 se repitió 50 veces, obteniendo valores medios del error RMS cometido en la estimación de la trayectoria del robot. En la Figura 7 representamos el error en la trayectoria estimada en función del número de vistas incluidas en el mapa y el tiempo necesario para calcular cada iteración. Típicamente, una aplicación de SLAM debe ser capaz de funcionar a tiempo real, por tanto, debe existir un compromiso entre la precisión del mapa y el tiempo de cálculo. Dada la capacidad de computación del robot, la Figura 7(a) nos permite determinar el número máximo  $N$  de vistas para poder procesar las observaciones a tiempo real y prever la precisión con la que podemos estimar la trayectoria del robot. Las Figuras 7(b) nos

permiten observar cómo varía el error RMS en función de las vistas del mapa. Obsérvese como el error tiende hacia un límite mínimo conforme  $N$  tiende a infinito. Por otra parte, en la Figura 7(c) se puede observar cómo aumenta el tiempo necesario de cómputo en función del error RMS deseado en el camino. Se puede comprobar cómo, el error no se corresponde de forma lineal con el tiempo de cálculo necesario.

## 5.2. SLAM: resultados con Datos Reales

En este apartado presentamos resultados que validan el esquema de SLAM propuesto mediante imágenes reales capturadas en un entorno interior. Los datos experimentales se obtuvieron con un robot Pioneer P3-AT equipado con una cámara firewire con una resolución  $1280 \times 960$  píxeles y un espejo hiperbólico. El eje óptico de la cámara está instalado aproximadamente perpendicular al plano del suelo como se describe en la Figura 1(a). Como consecuencia, una rotación del robot se corresponde con una rotación de la imagen respecto al eje óptico de la cámara. Durante las pruebas, se capturaron imágenes omnidireccionales cada vez que el robot avanzó más de  $0,05\text{ m}$  o giró más de  $0,05\text{ rad}$ . Igualmente, se almacenaron datos de distancia de un sensor SICK LMS y se obtuvo un mapa y un camino con el algoritmo descrito en (Stachniss et al., 2004) que se ayuda de la gran precisión de las medidas del sensor láser. Durante los experimentos el camino calculado a partir de datos de láser se utiliza únicamente para compararlos con los resultados de SLAM visual obtenidos mediante la cámara omnidireccional.

El robot es guiado a través del entorno mientras captura imágenes omnidireccionales y datos de distancia láser a lo largo de la trayectoria. De nuevo, para poder comparar resultados, hacemos uso de un algoritmo de SLAM basado en distancias

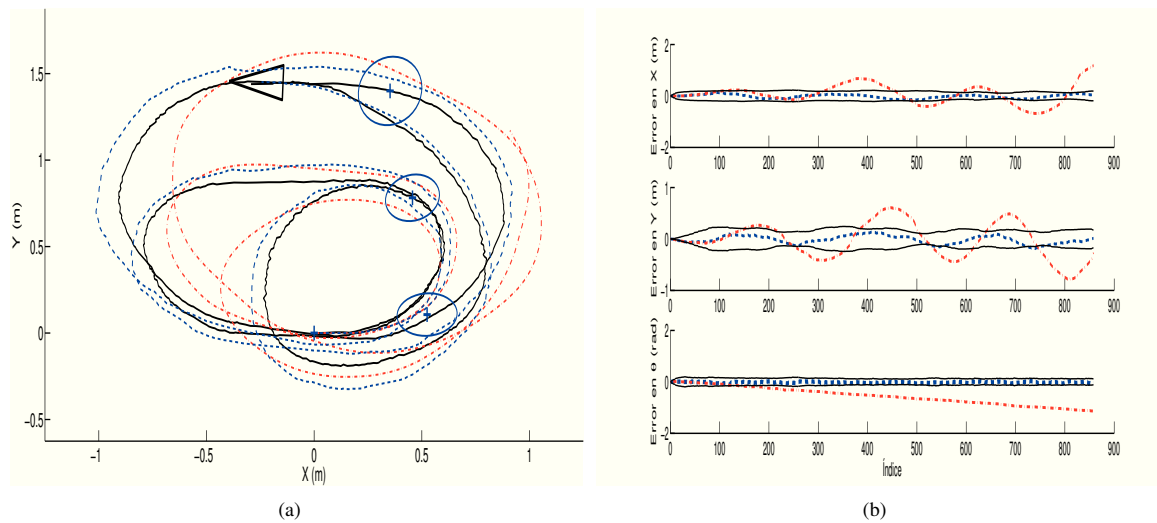


Figura 8: La Figura 8(a) presenta los resultados de SLAM con datos reales, para trayectoria real (punteada), estimación (continua) y odometría (trazos). La posición de las vistas se presenta con elipses de error. La Figura 8(b) presenta el error en cada paso temporal en  $X$ ,  $Y$  y  $\theta$  de la estimación (punteada) y la odometría (trazos).

láser, descrito en (Stachniss et al., 2004), para definir la trayectoria real. El robot comienza inicializando una vista a partir de la adquisición de una imagen omnidireccional en el origen. A continuación se mueve a lo largo de la trayectoria mientras continúa adquiriendo imágenes. Instantes después se inicializa una nueva vista. Mientras se calcula el mapa, se realiza una comparación entre la imagen actual y el resto de vistas del mapa, obteniendo un conjunto de correspondencias. Al mismo tiempo, la ratio de similitud (13) es evaluada, y cuando ésta cae por debajo de  $\delta_R = 0,5$ , se crea una nueva vista y se inicializa con la posición actual del robot. Finalmente el robot recorre la trayectoria mostrada en la Figura 10, donde mostramos con puntos el resto de posiciones en las que el robot decide inicializar una nueva imagen. La línea punteada muestra la trayectoria real, la línea continua muestra la estimación del EKF, mientras que la línea a trazos muestra la odometría. Cabe señalar que el robot continúa el movimiento dentro de la misma estancia siendo capaz de realizar observaciones de las vistas inicializadas anteriormente. En nuestro caso el umbral  $\delta_R$  fue determinado experimentalmente con el objetivo de generar un número reducido de vistas y representar el entorno de un modo más compacto. Si se eligiese un valor más bajo de  $\delta_R$ , menos imágenes serían inicializadas en el mapa. Por el contrario si se eligiese un valor superior, el mapa resultante almacenaría un mayor número de vistas. Puede observarse en la Figura 8(a) como una vez la cuarta vista es inicializada no es necesario inicializar ninguna otra, obteniendo así una representación más compacta. En la Figura 8(b) comparamos la trayectoria estimada con la trayectoria real y con la odometría. Hay que señalar que este error tiene la misma escala que la solución del mapa estimado y no ha sido normalizado. Presentamos el error en la estimación de la trayectoria (línea punteada) junto a los intervalos  $2\sigma$  y al error en la odometría (línea a trazos).

La Figura 9 presenta otro experimento. En este caso, el robot explora una habitación, recorre un pasillo, entra en una ha-

bitación diferente y vuelve al punto de origen. La distancia total recorrida es de  $45m$ . La Figura 9(a) presenta la trayectoria real (punteada), la odometría (a trazos) y la estimación (continua). La localización de las vistas y su incertidumbre asociada se indica mediante puntos y elipses de error. En la Figura 9(b) presentamos el error en la posición para cada paso temporal con intervalos de  $2\sigma$ . Puede observarse cómo el error presenta varias oscilaciones a lo largo del recorrido, lo cual se debe a momentos en los que el robot realiza giros muy pronunciados para entrar y salir de la habitación, así como para rodearla. Además, en estos instantes aparecen elementos obstruyentes y por tanto disminuye más si cabe la capacidad para visualizar vistas y obtener medidas de observación precisas. Este hecho puede comprobarse en la Figura 9(a), donde se observa cómo la disposición de las vistas del mapa en dichos instantes lleva asociada una mayor incertidumbre. Pese a todo sigue quedando de manifiesto que el filtro es capaz de mantener la convergencia en todo momento. Una vez que el robot vuelve a visualizar vistas almacenadas anteriormente, se comprueba que estos intervalos momentáneos de mayor incertidumbre se reducen, obteniendo un error para la estimación en torno a una decena de centímetros respecto al camino real.

Por último, la Figura 10 presenta otro experimento llevado a cabo en un entorno de dimensiones  $32 \times 45m$ , donde en la Figura 10(a) se representa la trayectoria real (continua), la odometría (a trazos) y la estimación (punteada). En este caso la distancia recorrida es mayor que en los casos anteriores y la complejidad del problema también aumenta debido a la presencia de un mayor número de obstrucciones. La localización de las vistas y su incertidumbre asociada se indica mediante puntos y elipses de error. En la Figura 10(b) presentamos el error en la posición para cada paso temporal con intervalos de  $2\sigma$ . Hay que destacar que los valores de error obtenidos para este caso son ligeramente mayores a causa de la complejidad del entorno. En general, los elementos obstruyentes hacen que existan muchos instantes

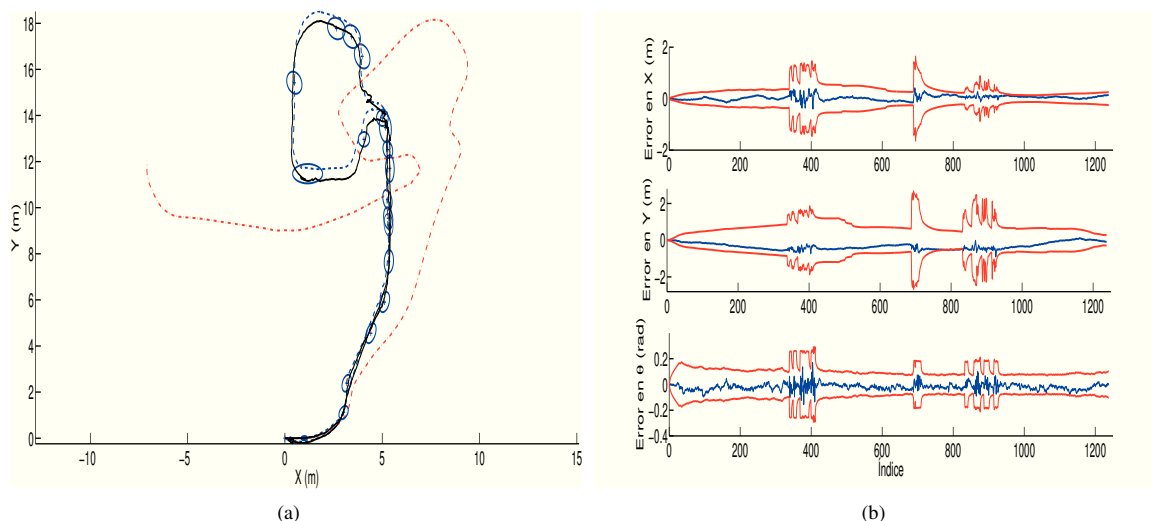


Figura 9: La Figura 9(a) presenta los resultados de SLAM con datos reales, para trayectoria real (punteada), estimación (continua) y odometría (trazos). La posición de las vistas se presenta con elipses de error. La Figura 9(b) presenta el error en cada paso temporal en  $X$ ,  $Y$  y  $\theta$  de la estimación (punteada) con intervalos de  $2\sigma$ .

en los cuales las observaciones se llevan a cabo con dificultad. Por ello la incertidumbre que se genera es mayor. Sin embargo, pese a esta situación desfavorable, el filtro EKF logra resolver el problema manteniendo la convergencia en todo momento, y asegurando un error dentro de los límites esperados.

## 6. Conclusiones

Hemos presentado un modelo para la resolución del problema de (SLAM) empleando imágenes omnidireccionales. Proponemos una representación del entorno que se aleja del concepto de mapa visual tradicional en el campo de SLAM visual. Habitualmente, el SLAM visual plantea la estimación de la posición 3D de un conjunto de marcas visuales y sus descriptores. En contraposición a este modelo, en este trabajo simplificamos el problema a la estimación de la posición y orientación de un conjunto reducido de imágenes omnidireccionales. Cada imagen omnidireccional, renombrada como *vista*, tiene asociado un conjunto de puntos de interés y sus descriptores visuales que describen el entorno de una forma compacta. Cada una de las imágenes permite representar un área del entorno, haciendo posible la localización del robot en las inmediaciones de cada una de ellas. La aportación fundamental se basa en la posibilidad de extraer una transformación entre dos imágenes omnidireccionales en las que existe un conjunto de correspondencias puntuales. Dicha transformación, formada por una rotación y una traslación (salvo un factor de escala), nos permite proponer un nuevo modelo de observación y resolver el problema de SLAM con un algoritmo basado en el EKF. Presentamos resultados obtenidos en entornos simulados que validan el esquema de SLAM en diferentes condiciones. Además, mostramos la validez de la propuesta con experimentos reales realizados con un robot móvil real. Las pruebas experimentales realizadas han demostrado que se puede modelar un entorno mediante un número reducido de imágenes omnidireccionales, lo que da lugar a un

problema con un menor número de variables a estimar. Al mismo tiempo, los resultados demuestran que este modelo permite obtener un buen resultado en términos de localización del robot, así como un mapa mucho más compacto que el obtenido con un mapa visual tradicional.

## English Summary

### Construction of a visual model of the environment based on omnidirectional images

#### Abstract

This paper deals with the problem of Simultaneous Localization and Mapping (SLAM). The solution presented is based on the utilisation of a set of images to represent the environment. In this way, the estimation of the map considers the computation of the position and orientation of a set of omnidirectional views captured from the environment. The proposed idea sets apart from the usual representation of a visual map, in which the environment is represented by a set of three dimensional points in a common reference system. Each of these points is commonly denoted as visual landmark. In the case presented here, the robot is equipped by a single omnidirectional visual sensor that allows to extract a number of interest points in the images, each one described by a visual descriptor. The map building process can be summed up in the following way: as the robot traverses the environment, it captures omnidirectional images and extracts a set of interest points from each one. Next, a set of correspondences is found between the current image and the rest of omnidirectional images existing in the map. When the number of correspondences found is enough, a transformation is computed, consisting of a rotation and a translation (up to an unknown scale factor). In the paper we show a method that allows to build a map while localizing the robot using these kind of observations. We present results obtained in a simula-

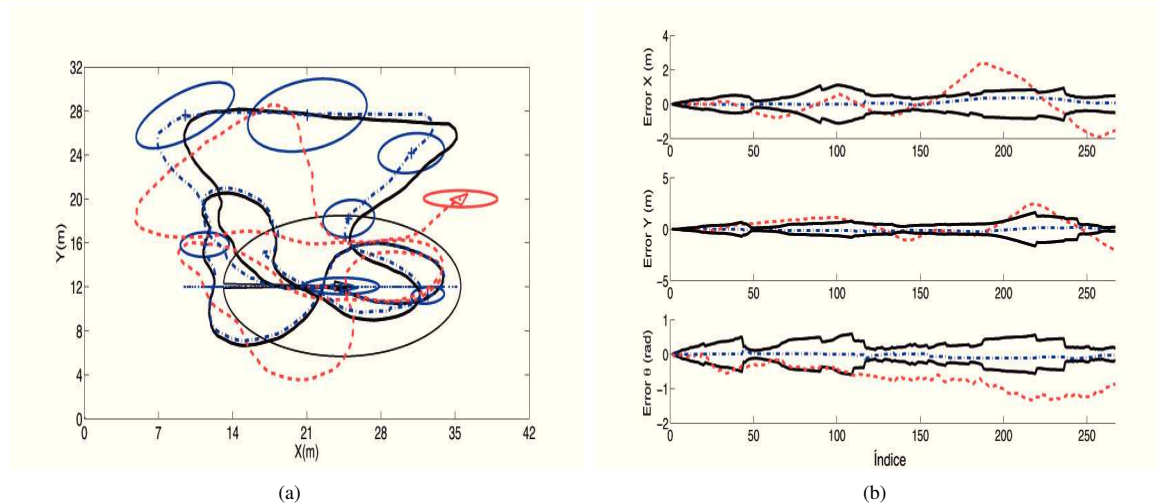


Figura 10: La Figura 10(a) presenta los resultados de SLAM con datos reales, para trayectoria real (continua), estimación (punteada) y odometría (trazos). La posición de las vistas se presenta con elipses de error. La Figura 10(b) presenta el error en cada paso temporal en  $X$ ,  $Y$  y  $\theta$  de la estimación (punteada) y la odometría (trazos) con intervalos de  $2\sigma$ .

ted environment that validate the proposed idea. In addition, we present experimental results using real data that prove the suitability of the solution.

#### Keywords:

SLAM, mobile robotics, omnidirectional vision

#### Agradecimientos

Este trabajo se ha llevado a cabo gracias en parte al Ministerio de Ciencia e Innovación a través del proyecto DPI2010-15308, con título "Exploración integrada de entornos mediante robots cooperativos para la creación de mapas 3D visuales y topológicos que puedan ser usados en navegación con 6 grados de libertad"

#### Referencias

- Andrew J. Davison, A. J., Gonzalez Cid, Y., Kita, N., 2004. Improving data association in vision-based SLAM. In: Proc. of IFAC/EURON. Lisboa, Portugal.
- Aracil, R., Balaguer, C., Armada, M., 2008. Robots de servicio. RIAI (Revista Iberoamericana de Automática e Informática Industrial) 5(2), 6–13.
- Ballesta, M., Gil, A., Reinoso, O., Úbeda, D., 2010. Análisis de detectores y descriptores de características visuales en slam en entornos interiores y exteriores. RIAI (Revista Iberoamericana de Automática e Informática Industrial) 7(2), 68–80.
- Bay, H., Tuytelaars, T., Van Gool, L., 2006. SURF: Speeded up robust features. In: Proc. of the ECCV. Graz, Austria.
- Bunschoten, R., Krose, B., 2003. Visual odometry from an omnidirectional vision system. In: Proc. of the ICRA.
- Civera, J., Davison, A. J., Martínez Montiel, J. M., 2008. Inverse depth parametrization for monocular slam. IEEE Trans. on Robotics.
- Davison, A. J., Murray, D. W., 2002. Simultaneous localisation and map-building using active vision. IEEE Trans. on PAMI.
- Gil, A., Martínez-Mozos, O., Ballesta, M., Reinoso, O., 2010. A comparative evaluation of interest point detectors and local descriptors for visual slam. Machine Vision and Applications.

- Gil, A., Reinoso, O., Martínez-Mozos, O., Stachniss, C., Burgard, W., 2006. Improving data association in vision-based SLAM. In: Proc. of the IROS. Beijing, China.
- Grisetti, G., Stachniss, C., Grzonka, S., Burgard, W., 2007. A tree parametrization for efficiently computing maximum likelihood maps using gradient descent. In: Proc. of RSS. Atlanta, Georgia.
- Harris, C. G., Stephens, M., 1988. A combined corner and edge detector. In: Proc. of Alvey Vision Conference. Manchester, UK.
- Hartley, R., Zisserman, A., 2004. Multiple View Geometry in Computer Vision. Cambridge University Press.
- Jae-Hean, K., Myung Jin, C., 2003. Slam with omni-directional stereo vision sensor. In: Proc. of the IROS. Las Vegas (Nevada).
- Joly, C., Rives, P., 2010. Bearing-only SAM using a minimal inverse depth parametrization. In: Proc. of ICINCO. Funchal, Madeira (Portugal).
- Kawanishi, R., Yamashita, A., Kaneko, T., 2008. Construction of 3D environment model from an omni-directional image sequence. In: Proc. of the Asia International Symposium on Mechatronics 2008. Sapporo, Japan.
- Lowe, D., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision.
- Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., 2002. Fastslam: a factored solution to the simultaneous localization and mapping problem. In: Proc. of the 18th national conference on Artificial Intelligence. Edmonton, Canada.
- Neira, J., Tardós, J. D., 2001. Data association in stochastic mapping using the joint compatibility test. IEEE Trans. on Robotics and Automation.
- Nister, D., 2003. An efficient solution to the five-point relative pose problem. In: Proc. of the IEEE CVPR. Madison, USA.
- Nistér, D., 2005. Preemptive RANSAC for live structure and motion estimation. Machine Vision and Applications.
- Paya, L., Fernández, L., Reinoso, O., Gil, A., Úbeda, D., 2009. Appearance-based dense maps creation: Comparison of compression techniques with panoramic images. In: 6th International Conference on Informatics in Control, Automation and Robotics ICINCO. Milan (Italy), pp. 250–255.
- Scaramuzza, D., Fraundorfer, F., Siegwart, R., 2009. Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In: Proc. of the ICRA. Kobe, Japan.
- Scaramuzza, D., Martinelli, A., Siegwart, R., 2006. A toolbox for easily calibrating omnidirectional cameras. In: Proc. of the IROS. Beijing, China.
- Stachniss, C., Grisetti, G., Haehnel, D., Burgard, W., 2004. Improved Rao-Blackwellized mapping by adaptive sampling and active loop-closure. In: Proc. of the SOAVE. Ilmenau, Germany.
- Stewenius, H., Engels, C., Nister, D., 2006. Recent developments on direct relative orientation. ISPRS Journal of Photogrammetry and Remote Sensing.