

# ACTAS

DE LAS

## XXXVIII Jornadas de Automática

Gijón · Palacio de Congresos · 6, 7 y 8 de Septiembre de 2017



Universidad de Oviedo  
*Universidá d'Uviéu*  
University of Oviedo



**CEA**  
Comité Español  
de Automática

Colabora

**Gijón**

Convention Bureau



Actas de

**XXXVIII**

**Jornadas de Automática**

© 2017 Universidad de Oviedo  
© Los autores

Servicio de Publicaciones de la Universidad de Oviedo  
Campus de Humanidades. Edificio de Servicios. 33011 Oviedo (Asturias)  
Tel. 985 10 95 03 Fax 985 10 95 07  
[http: www.uniovi.es/publicaciones](http://www.uniovi.es/publicaciones)  
[servipub@uniovi.es](mailto:servipub@uniovi.es)

DL AS 2749-2017

ISBN: 978-84-16664-74-0

Todos los derechos reservados. De conformidad con lo dispuesto en la legislación vigente, podrán ser castigados con penas de multa y privación de libertad quienes reproduzcan o plagien, en todo o en parte, una obra literaria, artística o científica, fijada en cualquier tipo y soporte, sin la preceptiva autorización.

# ALGORITMO DE SLAM UTILIZANDO APARIENCIA GLOBAL DE IMÁGENES OMNIDIRECCIONALES

Yerai Berenguer, Luis Payá, Mónica Ballesta, Luis M. Jiménez, Sergio Cebollada y Oscar Reinoso  
 Departamento de Ingeniería de Sistemas y Automática, Miguel Hernández University, Spain  
 {yberenguer, lpaya, m.ballesta, luis.jimenez, sergio.cebollada, o.reinoso}@umh.es

## Resumen

*Este trabajo presenta un algoritmo SLAM para estimar la posición y orientación de un robot móvil al mismo tiempo que crea un mapa del entorno. Utiliza solo la información visual proporcionada por un sistema catadióptrico montado en el robot, formado por una cámara que apunta hacia un espejo convexo, que proporciona al robot imágenes omnidireccionales que contienen información con un campo de visión de 360 grados alrededor del eje central del espejo. Cada imagen omnidireccional adquirida por el robot se describe utilizando descriptores de apariencia global, que permiten ejecutar nuestro algoritmo en tiempo real gracias a su compacidad. El método consta de tres pasos diferentes. En primer lugar, el robot calcula su posición y orientación, y crea un nuevo nodo en el mapa, el cual está formado por nodos conectados entre ellos. En segundo lugar, detecta cierres de bucle entre el nuevo nodo y los nodos del mapa. Finalmente, el mapa se optimiza utilizando un algoritmo de optimización y los cierres de bucle detectados. Se han utilizado dos conjuntos diferentes de imágenes para probar la eficacia del método, los cuales fueron capturados en dos entornos reales. Los resultados de los experimentos muestran la eficacia de nuestro método.*

**Palabras clave:** SLAM, Apariencia Global, Imágenes Omnidireccionales

## 1. INTRODUCCIÓN

El SLAM (Simultaneous Localization and Mapping) se ha considerado como una de las tecnologías más importantes en la investigación de la robótica móvil durante los últimos años [1, 2]. Muchos de estos trabajos se centran en el uso de información visual para desarrollar algoritmos de SLAM, debido a la rica información proporcionada por los sensores visuales.

Además de esto, los sensores visuales se pueden configurar de diferentes maneras como cámaras convencionales, sistemas estéreo, matrices de cámaras o sistemas catadióptricos. Esta última configuración está formada por una cámara apuntando hacia un espejo convexo [3]. La imagen resultante capturada por este último tipo de cámara contiene información sobre un campo de

visión de 360 grados alrededor del eje del espejo de la cámara. Esto proporciona imágenes omnidireccionales.

En el campo del SLAM, las imágenes omnidireccionales tienen multitud de ventajas gracias a su campo de visión, ya que una sola imagen contiene información de todos los alrededores del robot. Podemos encontrar muchos trabajos previos que usan imágenes omnidireccionales en tareas de creación de mapas y localización. Por ejemplo, Valiente et al. [4] presenta una comparación entre dos métodos visuales SLAM diferentes usando imágenes omnidireccionales y Garcia et al. [5] muestra una recopilación de métodos de creación de mapas y localización basados en visión por computador.

Tradicionalmente, las investigaciones sobre robots móviles que utilizan sensores visuales se basan en la extracción y descripción de algunos puntos característicos de las escenas, como descriptores SIFT (Scale-Invariant Feature Transform) [6] y SURF (Speeded-Up Robust Features) [7]. Este enfoque presenta algunas desventajas: el tiempo computacional para calcular y comparar los descriptores suele ser alto, y conduce a algoritmos de localización y creación de mapas relativamente complejos. Como ventaja, sólo se necesitan algunas posiciones almacenadas en el mapa para hacer posible el proceso de localización.

Más recientemente, algunos autores proponen utilizar la información global para describir las escenas, creando un descriptor único por imagen. Estas técnicas han demostrado ser una buena opción para resolver los problemas de localización y navegación cuando el movimiento del robot está contenido en el plano del suelo. Por ejemplo, Chang et al. [8] presenta un sistema de navegación y localización basado en la visión que utiliza el descriptor *gist*. Payá et al. [9] utilizan un descriptor basado en la firma de Fourier para realizar tareas de localización usando un algoritmo Monte Carlo, y Wu et al. [10] proponen un método de detección de cierre de bucle visual de manera eficiente. En [11], se analizan y comparan varios métodos para obtener descriptores globales de escenas panorámicas y se demuestra su validez en la construcción de mapas y la localización. La mayoría de estos descriptores de apariencia global se pueden usar en tiempo real porque el

tiempo computacional para calcularlos y manejarlos es bajo, y generalmente llevan a algoritmos de creación de mapas y localización más directos.

Habitualmente, el proceso de creación de mapas produce un error en cada posición del mapa debido al cálculo iterativo de nuevas poses (posición y orientación) del robot. Esto puede ser un gran problema en entornos extensos cuando el robot tiene que calcular muchas poses nuevas, ya que el error está aumentando en cada iteración. Esta incertidumbre puede reducirse mediante la detección de cierres de bucle y el uso de algoritmos de optimización para reubicar las poses. Este problema se estudia a fondo en este trabajo.

La contribución de este trabajo es la creación de un método para llevar a cabo las tareas de SLAM utilizando sólo la información visual del entorno y los descriptores de apariencia global. Cada escena omnidireccional adquirida por el robot se describe utilizando estos descriptores. El método consta de tres pasos diferentes: calcular la pose del robot (posición y orientación), detectar cierres de bucle (comparando descriptores de apariencia global) y optimizar el mapa (utilizando el algoritmo de optimización G2O). El algoritmo de optimización utilizado se denomina G2O y fue presentado por Kümmerle et al. [12].

Los experimentos se han llevado a cabo con dos conjuntos diferentes de imágenes capturadas en dos entornos reales de trabajo. El primero se ha tomado siguiendo un camino rectangular en un entorno interior y el segundo ha sido capturado siguiendo un camino real incluyendo varias habitaciones en un edificio.

El resto de este trabajo se estructura de la siguiente manera. La sección 2 introduce algunos conceptos preliminares sobre la descripción de la imagen y la optimización de grafos. La sección 3 presenta el algoritmo de SLAM que hemos implementado para resolver el problema de localización y creación de mapas de manera simultánea. La sección 4 describe las bases de datos utilizadas para llevar a cabo los experimentos y presenta los resultados obtenidos. Finalmente, la sección 5 esboza las conclusiones.

## 2. PRELIMINARES

A lo largo de este artículo, se utilizan dos métodos para describir la apariencia global de las escenas: la transformada de Radon y HOG (Histogram of Oriented Gradients), los cuales son descritos en esta sección. Además, se presentan los fundamentos de los métodos utilizados para calcular la diferencia entre dos imágenes capturadas en diferentes posiciones. Por último, describimos el algoritmo de optimización utilizado para recalcular las posiciones de mapa después de detectar cierres de bucle.

### 2.1. Descriptores de Apariencia Global

Los métodos basados en la apariencia global de las escenas constituyen una alternativa robusta a los métodos basados en la extracción de características. Esto se debe a que los descriptores basados en apariencia global representan el entorno a través de características de alto nivel que se pueden interpretar y manejar fácilmente.

Esta subsección presenta los descriptores que hemos utilizado para describir las imágenes omnidireccionales. Ambos se basan en la apariencia global, sin ninguna segmentación o extracción local de características.

#### 2.1.1. Transformada de Radon

La transformada de Radon se describió inicialmente en [13]. Existen trabajos que demuestran la eficacia de este descriptor en la descripción de formas y en la segmentación, tales como [14] y [15]. Hoang et al. [14] presentan un descriptor de forma, invariante a transformaciones geométricas, basado en las transformadas de Radon, Fourier y Mellin, y Hasegawa et al. [15] presentan un descriptor de formas que combina el histograma de la transformada de Radon y la correlación de fase. Berenguer et al. [16] presentan un método de localización 2D usando un descriptor de apariencia global basado en la transformada Radon y demuestran la eficacia y robustez de este descriptor.

Matemáticamente, la transformada de Radón de una imagen  $im(i, j) \in \mathbb{R}^{K \times L}$  a lo largo de la línea  $c_1(\phi, d)$  (Figura 1) puede obtenerse como:

$$\begin{aligned} \mathcal{R}\{im(i, j)\} &= \lambda_f(\phi, d) = \\ &= \int_{\mathbb{R}} im(d \cos \phi - j' \sin \phi, d \sin \phi + j' \cos \phi) dj' \quad (1) \end{aligned}$$

donde  $\mathcal{R}$  es el operador de la transformada de Radon.  $im(i, j)$  es la imagen a transformar.  $\lambda_f$  es la imagen transformada, la cual depende de dos variables diferentes: la distancia desde la línea  $c_1$  hasta el origen  $d$  y del ángulo entre los ejes  $x$  e  $i'$ ,  $\phi$  (Figura 1). El eje  $j'$  es paralelo a la línea  $c_1$ .

Si consideramos diferentes valores para  $d$  y  $\phi$  en la Ecuación (1), la función transformada  $\lambda_f(\phi, d)$  se convertirá en una matriz con  $M$  filas y  $N$  columnas.  $M$  es el número de orientaciones consideradas (normalmente elegidas para cubrir toda la circunferencia), y  $N$  es el número de líneas paralelas consideradas en cada orientación (para cubrir toda la imagen). La distancia entre cada par de líneas consecutivas se considera constante.

Cuando la transformada de Radon se aplica a una imagen, se calculan las proyecciones de la imagen a lo largo de las direcciones especificadas a través de un grupo de integrales de línea a lo largo de un haz de

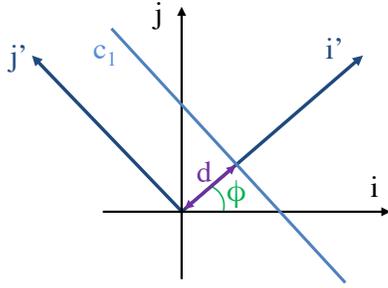


Figura 1: Parametrización de línea mediante la distancia origen  $d$  y el ángulo entre la línea normal y el eje  $i$ ,  $\phi$ .

líneas paralelas a dichas direcciones. La distancia entre cada par de líneas paralelas suele ser un píxel.

### 2.1.2. Histogram of Oriented Gradients (HOG)

HOG se ha utilizado tradicionalmente como un método de descripción en el campo de la detección de objetos. Fue descrito inicialmente por [17] y lo usaron en tareas de detección de personas. Sin embargo, hay varias investigaciones en las que se ha mejorado este método de descripción, como [18], donde mejoran la precisión y el coste computacional.

La implementación básica consiste en dividir la imagen en pequeñas celdas conectadas y calcular el histograma de las orientaciones del gradiente en cada celda. Entonces, el descriptor se compone de estos histogramas dispuestos en un solo vector.

Fernández et al. [19] analizan este tipo de descriptor en tareas de localización al aire libre. Además, hacen un análisis comparativo entre varios métodos para describir imágenes panorámicas.

### 2.2. Phase Only Correlation (POC)

En esta subsección se presenta el método utilizado para comparar la transformada de Radon de dos imágenes, el cual proporciona una medida de la diferencia entre la apariencia visual de dos ubicaciones y la estimación del cambio de orientación del robot entre estas ubicaciones.

POC (Phase Only Correlation), propuesto en [20], es una operación realizada en el dominio de la frecuencia que proporciona un coeficiente de correlación entre dos imágenes [21]. En nuestro caso, comparamos dos transformadas de Radon, pero esto no afecta al funcionamiento de POC porque la transformada de Radon se puede interpretar como una imagen. En general, permite obtener tanto la orientación relativa entre dos diferentes transformadas de Radon como un coeficiente de similitud entre ellas, como se muestra en [16].

La correspondencia entre dos imágenes  $im_1(i, j)$  e  $im_2(i, j)$  calculadas por POC viene dada por la siguiente ecuación:

$$C(i, j) = \mathcal{F}^{-1} \left\{ \frac{\mathbf{IM}_1(u, v) \cdot \mathbf{IM}_2^*(u, v)}{|\mathbf{IM}_1(u, v) \cdot \mathbf{IM}_2^*(u, v)|} \right\} \quad (2)$$

Donde  $\mathbf{IM}_1$  es la transformada de Fourier de la imagen 1 y  $\mathbf{IM}_2^*$  es el conjugado de la transformada de Fourier de la imagen 2.  $\mathcal{F}^{-1}$  es el operador inverso de la transformada de Fourier.

Para estimar la distancia entre las dos imágenes ( $im_1$  e  $im_2$ ) hemos utilizado la siguiente expresión:

$$dist(im_1, im_2) = 1 - \max\{C(i, j)\} \quad (3)$$

$\max\{C(i, j)\}$  es un coeficiente que toma valores en el intervalo  $[0, 1]$  y mide la similitud entre las dos imágenes  $im_1$  e  $im_2$ .

Esta operación es invariante frente a desplazamientos de las imágenes a lo largo de los ejes  $i$  y  $j$ . Además, es posible estimar estos desplazamientos  $\Delta_x$  y  $\Delta_y$  a lo largo de ambos ejes mediante:

$$(\Delta_x, \Delta_y) = \operatorname{argmax}_{(i, j)} \{C(i, j)\} \quad (4)$$

Si comparamos las transformadas de Radon de dos imágenes omnidireccionales usando POC, el valor  $\Delta_x$  es proporcional a la orientación relativa  $\alpha$  del robot al capturar las imágenes, de acuerdo con la Ecuación (5). La Figura 2 muestra las transformaciones de Radon de dos imágenes omnidireccionales diferentes capturadas desde el mismo punto  $(x_w, y_w, z_w)$  pero con orientaciones del robot diferentes con respecto al eje  $z_w$ ,  $\theta$  (Figura 2).

$$\alpha = \frac{\Delta_x \cdot 2\pi}{N} \quad (5)$$

De esta manera, POC es capaz de comparar dos imágenes independientemente en la orientación y también es capaz de estimar este cambio de orientación.

### 2.3. Algoritmo de optimización: G2O

G2O es un algoritmo de optimización descrito en [12]. Este método fue creado para optimizar funciones de errores no lineales basadas en grafos.

En el campo del SLAM, el robot tiene que calcular su pose cuando toma cada nueva imagen con respecto a las poses anteriores almacenadas en el mapa. Esta operación tiene un error asociado que va incrementando con cada nuevo cálculo de pose, por lo que necesitamos corregir las poses almacenadas en el mapa para disminuir esta desviación. G2O puede volver a calcular cada pose del mapa usando nuevas restricciones.

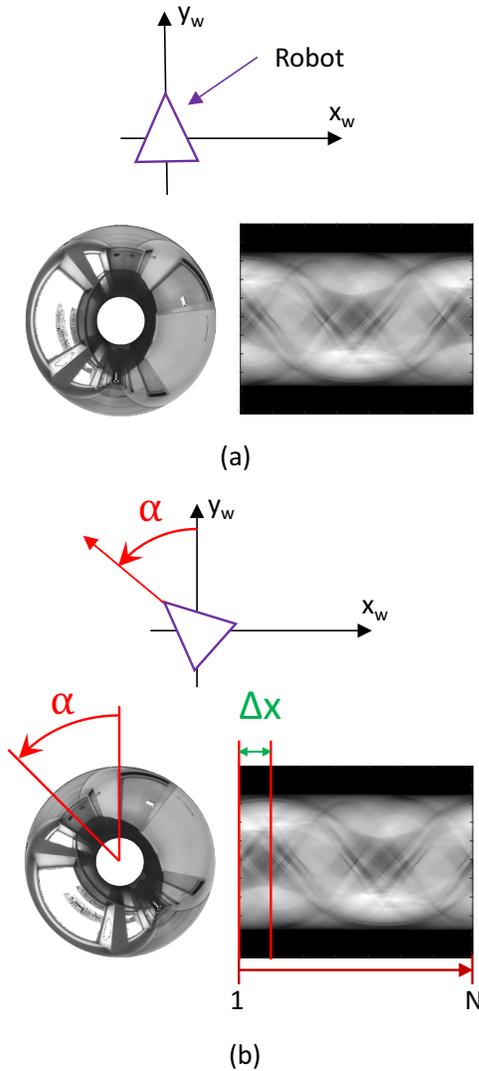


Figura 2: (a) Imagen omnidireccional capturada desde una posición específica de un entorno virtual y su transformada Radon. (b) Imagen omnidireccional tomada desde el mismo lugar cambiando solamente la orientación del robot alrededor del eje  $z_w$ , y su transformada Radon. Un cambio en la orientación del robot alrededor del eje  $z_w$  produce un desplazamiento en las columnas de la transformada Radon,  $\Delta x$ .

Una de estas restricciones se puede obtener cuando se producen cierres de bucle entre una pose del mapa existente y la nueva pose del robot. Entonces, G2O reubica cada nodo del mapa modificándolos gradualmente para cumplir con la restricción de cierre de bucle. Entonces, el nuevo nodo se ubica en la misma posición que la pose equivalente almacenada en el mapa.

### 3. ALGORITMO DE SLAM

En esta sección, presentamos nuestro algoritmo de SLAM visual. El robot pasa por el entorno y captu-

ra imágenes de algunas posiciones. Cada vez que llega una nueva imagen, el robot incluye un nuevo nodo dentro del mapa, el cual está formado por nodos. A continuación, se resuelve el problema SLAM siguiendo estos tres pasos:

Primero, el robot calcula dos descriptores de la imagen: la transformada de Radon y el descriptor HOG y los almacena en el nodo. Entonces, el robot crea un nuevo nodo y lo ubica dentro del mapa calculando la posición y orientación del nuevo nodo con respecto al nodo previamente agregado y se crea un enlace entre ambos nodos. Este proceso de localización se realiza utilizando sólo información visual.

En segundo lugar, el robot comprueba la existencia de posibles cierres de bucle comparando la nueva escena con las escenas anteriores almacenadas en el mapa.

Finalmente, el mapa se optimiza utilizando el algoritmo G2O con los cierres de bucle detectados. Este proceso se repite en cada nueva ubicación.

#### 3.1. Creación del mapa

Esta subsección presenta el método propuesto para calcular las coordenadas de las poses de los nuevos nodos. Estas poses se calculan obteniendo la distancia y el ángulo entre las escenas. Para cada nuevo nodo, el robot almacena el descriptor de Radon y el descriptor HOG de la nueva imagen omnidireccional para hacer posible su localización.

La Figura 3 muestra un esquema del proceso de creación de mapas. Consiste en el cálculo de las coordenadas  $(x_k, y_k)$  de cada nuevo nodo. Estas coordenadas se calculan a partir de la distancia y el ángulo respecto a la posición anterior.

La distancia entre ubicaciones se calcula usando la Ecuación 3. Es una distancia imagen y no es una distancia métrica, es decir, esta distancia no es una unidad de medida real, sino que es proporcional a la distancia métrica.

$\beta_k$ , es aproximadamente el ángulo de orientación,  $\alpha_k$ , calculado usando la Ecuación 5:

$$\beta_k \approx \alpha_k \quad (6)$$

Cuanto más cerca están las poses, más exacta es la aproximación, ya que el cambio de orientación del robot es menor, y el error cometido en la aproximación se reduce. Sin embargo, cabe tener en cuenta que se trata de estimaciones cuyo error se reducirá durante el posterior proceso de optimización cuando se detecte un cierre de ciclo.

Finalmente, las coordenadas de cada nodo  $(x_k, y_k)$  se calculan mediante estas ecuaciones:

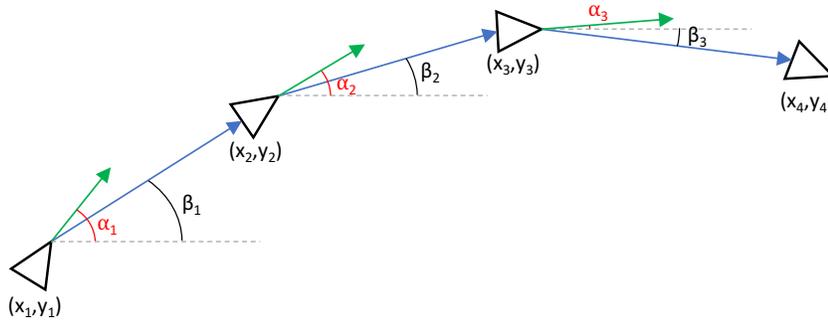


Figura 3: Esquema del proceso de creación de mapas.

$$x_k = \text{dist}(im_{k-1}, im_k) \cdot \cos(\alpha_k) \quad (7)$$

$$y_k = \text{dist}(im_{k-1}, im_k) \cdot \sin(\alpha_k) \quad (8)$$

donde  $\text{dist}(im_{k-1}, im_k)$  es la distancia POC entre dos imágenes consecutivas calculadas usando la Ecuación 3, y  $\alpha_k$  es el ángulo de orientación del nodo  $k$ , calculado usando la Ecuación 5.

### 3.2. Cierres de Bucle

El siguiente paso del algoritmo consiste en detectar cierres de bucle. Para ellos, se compara el descriptor HOG de la nueva imagen tomada por el robot con los descriptores HOG almacenados en el mapa. Para calcular la distancia entre los descriptores HOG se utiliza la similitud de coseno entre ellos para calcular la distancia:

$$\text{dist}(\vec{d}_1, \vec{d}_2) = 1 - \frac{\vec{d}_1 \cdot \vec{d}_2^T}{\sqrt{(\vec{d}_1 \cdot \vec{d}_1^T)(\vec{d}_2 \cdot \vec{d}_2^T)}} \quad (9)$$

donde  $\vec{d}_1$  y  $\vec{d}_2$  son los descriptores HOG de dos imágenes diferentes.

Los cierres de bucle deben determinarse definiendo un umbral máximo de distancia,  $W$  (Ecuación 10). Este umbral se define como una constante en el inicio del proceso de SLAM. Si la distancia es inferior a este umbral, las dos poses comparadas se considerarán como la misma posición  $(x, y)$ , pero la orientación del robot puede ser diferente.

$$\text{if}(\text{dist}(\vec{d}_1, \vec{d}_2) < W) \rightarrow \text{cierre de bucle} \quad (10)$$

### 3.3. Optimización del mapa

Teniendo en cuenta los cierres de bucle detectados, el robot utiliza esta información para optimizar el mapa almacenado. Esta optimización se realiza utilizando el algoritmo de optimización G2O.

Cuando el robot detecta un cierre de bucle, tiene que reubicar todos los nodos anteriores para reducir el error asociado en cada posición. Este proceso modifica todas las posiciones de los nodos en el mapa para tener en cuenta la nueva restricción calculada por la detección de cierre de bucle.

La modificación de la ubicación de los nodos se realiza mediante el algoritmo G2O. Dicho algoritmo recibe como entrada todas las posiciones de los nodos del mapa y la restricción de cierre de bucle. Entonces, G2O da como salida las nuevas posiciones de los nodos recalculadas.

Por lo tanto, los dos nodos del cierre del bucle se localizan en la misma posición y se modifican las coordenadas del resto de los nodos del mapa.

## 4. EXPERIMENTOS

Esta subsección presenta los diferentes conjuntos de imágenes omnidireccionales utilizados para probar nuestro método y los resultados obtenidos en estos experimentos.

### 4.1. Bases de datos

Para comprobar el funcionamiento de la técnica propuesta, se utilizan dos conjuntos de imágenes capturadas por nosotros mismos. Para capturar el primer conjunto, el robot fue teleoperado para seguir un camino rectangular. El segundo conjunto de imágenes fue capturado mientras el robot seguía un camino más complicado a través de varias habitaciones dentro de un edificio. Figura 4 muestra una imagen omnidireccional de muestra de cada entorno.

Estas dos bases de datos se han creado tomando una imagen omnidireccional cada 40 cm aproximadamente. La Figura 5 muestra el sistema catadióptrico utilizado para capturar las imágenes omnidireccionales, formado por la cámara (modelo: DFK-41BF02) y el espejo hiperbólico (modelo: Eizo Wide70).

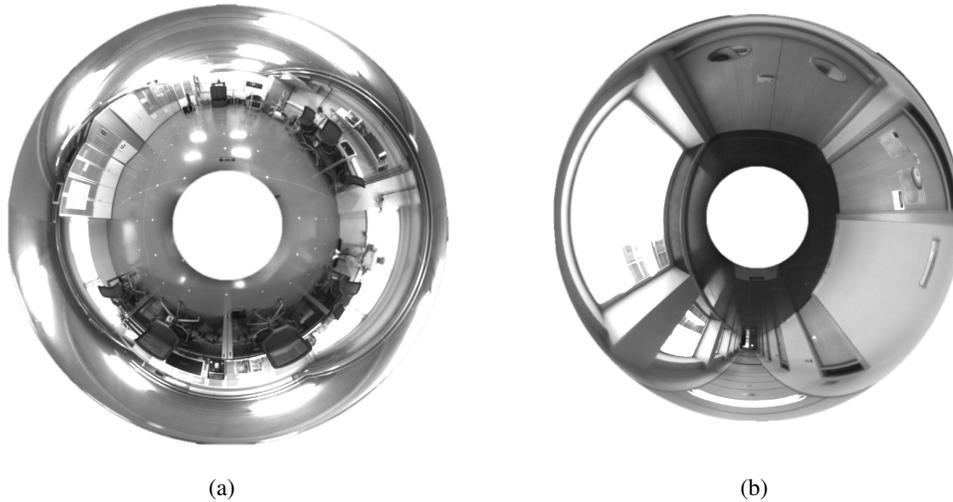


Figura 4: (a) Muestra de una imagen omnidireccional de la trayectoria rectangular.(b) Imagen de muestra del segundo recorrido.



Figura 5: Sistema de adquisición de imágenes omnidireccionales.

## 4.2. Resultados

En esta sección se muestran los resultados de los experimentos realizados con nuestro algoritmo SLAM. Las dos bases de datos descritas en la sección 4.1 se han utilizado para llevar a cabo estos experimentos.

El umbral máximo de distancia entre descriptores HOG es un parámetro importante a sintonizar. Para ello, hemos hecho algunas pruebas y elegido el mejor valor para detectar cierres de bucles. Después de estas pruebas se considera un umbral igual a 0,006 como un buen valor de la distancia entre descriptores HOG.

La Figura 6 muestra los resultados del algoritmo SLAM después de incorporar la posición final de la

primera ruta. La línea azul es el mapa creado sin optimización y la línea verde es el mismo mapa optimizado. Esta optimización se realiza en cada iteración pero el mapa sin ninguna optimización se muestra a efectos comparativos. Como podemos ver, la línea verde es un camino cuadrado.

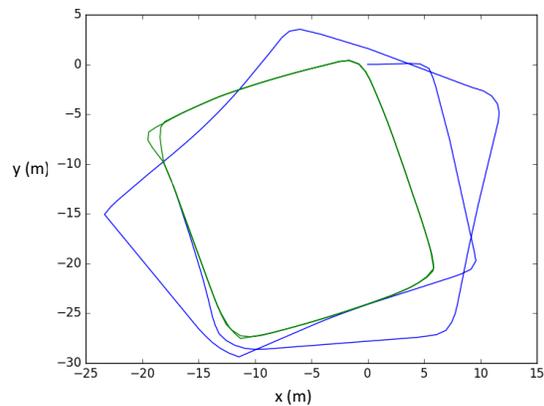


Figura 6: Mapa creado utilizando el primer recorrido. La línea azul es el mapa creado sin optimización y la línea verde es el mismo mapa optimizado.

La Figura 7 muestra los mismos resultados que en la Figura 6 pero utilizando la segunda ruta. La línea azul es el mapa creado sin optimización, la línea verde es el mismo mapa optimizado y la línea amarilla es la trayectoria real.

En cuanto al tiempo computacional, el robot pasa un promedio de 0,65 segundos en cada iteración del proceso SLAM. Este tiempo aumenta en cada iteración porque el mapa está formado por mayor cantidad de nodos y la detección de cierre de bucle necesita comparar un mayor número de descriptores HOG.

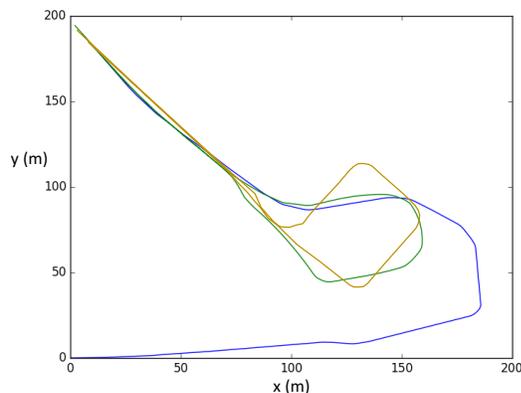


Figura 7: Mapa creado utilizando el segundo recorrido. La línea azul es el mapa creado sin optimización, la línea verde es el mismo mapa optimizado y la línea amarilla es la trayectoria real.

## 5. CONCLUSIONES

En este trabajo hemos presentado un método SLAM para estimar la posición y la orientación de un robot móvil en un entorno a la vez que el robot crea el mapa. Utilizamos dos descriptores de apariencia global diferentes para llevar a cabo el proceso SLAM y el mapa está formado por estos dos descriptores de cada imagen. Por último, el algoritmo ha sido probado con dos conjuntos de imágenes capturadas en dos entornos interiores diferentes.

Los resultados han demostrado la exactitud del método. En cuanto a los valores de los parámetros, el umbral máximo de distancia entre descriptores HOG es el parámetro de ajuste principal en este método.

Los resultados presentados en este trabajo muestran la efectividad de los descriptores de apariencia global de imágenes omnidireccionales para llevar a cabo tareas de SLAM gracias a la riqueza de la información que contienen. Ahora estamos trabajando para mejorar este método y estamos tratando de estimar con mayor precisión la orientación relativa entre los nodos. Además, estamos implementando un método de agrupación para reducir el tiempo computacional para detectar cierres de bucle cuando se aumenta el número de nodos.

## AGRADECIMIENTOS

Este trabajo ha sido apoyado por el gobierno español a través del proyecto DPI2016-78361-R (AEI/FEDER, UE) “Creación de Mapas Mediante Métodos de Apariencia Visual para la Navegación de Robots”.

## Referencias

- [1] R. Munguia, S. Urzua, and A. Grau, “Delayed monocular slam approach applied to unmanned aerial vehicles,” *PLOS ONE*, vol. 11, no. 12, pp. 1–24, 12 2016.
- [2] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, “Elasticfusion: Real-time dense slam and light source estimation,” *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.
- [3] H. Nagahara, K. Yoshida, and M. Yachida, “An omnidirectional vision sensor with single view and constant resolution,” in *2007 IEEE 11th International Conference on Computer Vision*, Oct 2007, pp. 1–8.
- [4] D. Valiente, A. Gil, L. Fernández, and O. Reinoso, “A comparison of ekf and sgd applied to a view-based slam approach with omnidirectional images,” *Robotics and Autonomous Systems*, vol. 62, no. 2, pp. 108–119, 2014.
- [5] E. Garcia-Fidalgo and A. Ortiz, “Vision-based topological mapping and localization methods: A survey,” *Robotics and Autonomous Systems*, vol. 64, pp. 1–20, 2015.
- [6] D. Lowe, “Object recognition from local scale-invariant features,” in *ICCV 1999, International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.
- [7] H. Bay, T. Tuytelaars, and L. Gool, “Surf: Speeded up robust features,” *Computer Vision at ECCV*, vol. 3951, pp. 404–417, 2006.
- [8] C. Chang, C. Siagian, and L. Itti, “Mobile robot vision navigation and localization using gist and saliency,” in *IROS 2010, International Conference on Intelligent Robots and Systems*, 2010, pp. 4147–4154.
- [9] L. Payá, L. Fernández, L. Gil, and O. Reinoso, “Map building and monte carlo localization using global appearance of omnidirectional images,” *Sensors*, vol. 10, no. 12, pp. 11 468–11 497, 2010.
- [10] J. Wu, H. Zhang, and Y. Guan, “An efficient visual loop closure detection method in a map of 20 million key locations,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 861–866.
- [11] L. Payá, F. Amorós, L. Fernández, and O. Reinoso, “Performance of global-appearance descriptors in map building and localization using omnidirectional vision,” *Sensors*, vol. 14, no. 2, pp. 3033–3064, 2014.
- [12] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G2o: A general framework for graph optimization,” in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3607–3613.
- [13] J. Radon, “Über die bestimmung von funktionen durch ihre integralwerte langs gewisser mannigfaltigkeiten,” *Berichte Sachsische Akademie der Wissenschaften*, vol. 69, no. 1, pp. 262–277, 1917.
- [14] T. Hoang and S. Tabbone, “A geometric invariant shape descriptor based on the radon, fourier, and mellin transforms,” in *20th International Conference on Pattern Recognition (ICPR)*, Aug 2010, pp. 2085–2088.

- [15] M. Hasegawa and S. Tabbone, "A shape descriptor combining logarithmic-scale histogram of radon transform and phase-only correlation function," in *2011 International Conference on Document Analysis and Recognition (ICDAR)*, Sept 2011, pp. 182–186.
- [16] Y. Berenguer, L. Payá, M. Ballesta, and O. Reinoso, "Position estimation and local mapping using omnidirectional images and global appearance descriptors," *Sensors*, vol. 15, no. 10, p. 26368, 2015.
- [17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, June 2005, pp. 886–893 vol. 1.
- [18] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 1491–1498.
- [19] L. Fernández, L. Payá, O. Reinoso, L. Jiménez, and M. Ballesta, "A study of visual descriptors for outdoor navigation using google street view images," *Journal of Sensors*, vol. 2016, 2016.
- [20] C. Kuglin and D. Hines, "The phase correlation image alignment method," in *In Proceedings of the IEEE, International Conference on Cybernetics and Society*, 1975, pp. 163–165.
- [21] K. Kobayashi, T. Aoki, K. Ito, H. Nakajima, and T. Higuchi, "A fingerprint matching algorithm using phase-only correlation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, pp. 682–691, 2004.