

# CREACIÓN DE MAPAS DENSOS BASADOS EN APARIENCIA

Lorenzo Fernández, Luis Payá, David Úbeda, Mónica Ballesta, Jose M. Marín

Departamento de Ingeniería de Sistemas Industriales.  
Universidad Miguel Hernández. Avda. de la Universidad s/n. 03202, Elche (Alicante), Spain  
l.fernandez@umh.es, lpaya@umh.es

## Resumen

*La información visual capturada por los sistemas omnidireccionales es muy rica y puede ser muy útil en aplicaciones en las que un robot debe crear un mapa de un entorno. Este mapa podría estar compuesto por varias imágenes panorámicas tomadas desde diferentes puntos de vista en el entorno, añadiendo diversas relaciones geométricas entre ellas. Para llevar a cabo cualquier tarea, habitualmente es necesario que el robot calcule su posición y orientación en el entorno, comparando su información visual actual con los datos almacenados en el mapa. En este trabajo se estudian y se comparan algunos enfoques para crear el mapa, utilizando métodos basados en apariencia. Un factor muy a tener en cuenta en estos enfoques es el tipo de información a almacenar con el fin de minimizar el coste computacional de las operaciones necesarias. Para ello, hemos llevado a cabo una experimentación exhaustiva, que ha permitido estudiar la cantidad de memoria que requieren varias técnicas para construir el mapa, y el tiempo necesario para crearlo y para llevar a cabo el proceso de localización conjuntamente. Además, se ha analizado la precisión para calcular la posición y la orientación de un robot en el entorno.*

**Palabras Clave:** Construcción de mapas, métodos basados en apariencia, visión omnidireccional, localización espacial.

## 1 INTRODUCCIÓN

Cuando un robot o un equipo de robots ha de llevar a cabo una tarea en un entorno, por lo general, es necesaria una representación interna del mismo, de modo que el robot pueda estimar su posición y orientación inicial y navegar hacia los puntos de destino. Los sistemas de visión omnidireccional se usan habitualmente con este objetivo, debido a su bajo coste y a la cantidad de información que proporcionan. La mayor parte de las investigaciones llevadas a cabo en este ámbito se han centrado en la extracción de marcas naturales o artificiales de las

imágenes para contruir el mapa y llevar a cabo la localización del robot [9]. Sin embargo, este proceso puede realizarse trabajando con las imágenes en su conjunto, sin necesidad de extraer marcas ni regiones características. Estos enfoques basados en apariencia son útiles cuando trabajamos en medios no estructurados donde puede ser difícil la creación de modelos apropiados de reconocimiento, y ofrecen un método sistemático e intuitivo para construir el mapa. No obstante, como no se extrae información relevante de las imágenes, un problema importante de tales enfoques es el alto coste computacional que suponen.

Diferentes investigadores han mostrado como puede ser usada una representación del entorno en un subespacio de menor orden, usando técnicas de compresión. Un método ampliamente extendido es PCA (Principal Components Analysis). Como ejemplo, en [5] se crea una base de datos usando un conjunto de escenas con un enfoque probabilístico para la localización dentro de la base de datos. Sin embargo, los métodos convencionales de PCA no aprovechan la cantidad de información que ofrecen las cámaras omnidireccionales, porque no pueden tratar con rotaciones en el plano en que se mueve el robot. Uenohara [10] estudió ese problema con un conjunto de imágenes rotadas, y Jogan [3] aplicó estos conceptos a un mapa basado en apariencia de un entono. El enfoque consiste en la creación de un espacio propio que tenga en cuenta las posibles rotaciones de cada imagen de entrenamiento, intentando llegar a un compromiso entre la cantidad de memoria, tiempo y precisión del mapa. Otros trabajos usan la Transformada de Fourier Discreta para comprimir la información, como Menegatti [6], que define el concepto de Firma de Fourier y presenta un método para construir el mapa y localizar al robot dentro del mismo, o Rossi [8], que usa la transformada de Fourier esférica de las imágenes omnidireccionales.

La representación de un entorno con métodos basados en apariencia puede ser separada en un mapa de bajo nivel, que representa una habitación con varias imágenes y un mapa de alto nivel, que intenta modelizar las relaciones espaciales entre habitaciones y entre habitaciones y pasillos. Booi [2] muestra

como estos conceptos pueden implementarse mediante un grafo cuyos nodos son las imágenes y cuyos enlaces denotan la similitud entre ellos y Vasudevan [11] usa una jerarquía de mapas cognitivos donde las celdas representan las escenas mediante una compresión PCA, y las relaciones de conexión entre ellas se obtienen a partir de la información suministrada por una brújula electrónica. Las técnicas basadas en apariencia constituyen un modelo de referencia para otras aplicaciones en robótica, como en seguimiento de rutas, como muestra Payá [7].

## 2 ANÁLISIS DE TÉCNICAS DE COMPRESIÓN APLICADAS A IMÁGENES PANORÁMICAS

En esta sección se expondrán, en términos generales, algunas técnicas para extraer la información más relevante de un conjunto de imágenes panorámicas, capturadas desde varias posiciones dentro del entorno del que se pretende crear el mapa.

### 2.1 TÉCNICAS BASADAS EN PCA

Quando tomamos un conjunto de  $N$  imágenes con  $M$  píxeles cada una,  $\vec{x}^j \in \mathfrak{R}^{M \times 1}; j=1 \dots N$ , cada imagen puede ser transformada en un vector de características (también llamado ‘proyección’ de la imagen)  $\vec{p}^j \in \mathfrak{R}^{K \times 1}; j=1 \dots N$ , siendo  $K$  las características PCA que contienen la información más relevante de la imagen,  $K \leq N$  [4]. La transformada PCA se puede calcular mediante la descomposición en valores singulares de la matriz de covarianza  $C$  de la matriz de datos  $X$ , que contiene todo el conjunto de imágenes ordenadas en columnas (sustrayendo la media a cada una de ellas). Si  $V$  es la matriz que contiene los  $K$  principales vectores propios y  $P$  es la matriz de proyecciones, de tamaño  $K \times N$ , la reducción de dimensionalidad se hace mediante la operación  $P = V^T \cdot X$ , donde las columnas de  $P$  son las proyecciones del conjunto de imágenes.

Sin embargo, la construcción de la base de datos de esta forma contiene información sólo de la orientación que el robot tenía cuando se capturó cada imagen pero no para todas las posibles orientaciones en cada punto. Jogan [3] presenta una metodología que incluye dicha información de orientación, con la particularidad de que sólo es necesario adquirir una imagen por posición. Cuando trabajamos con una imagen panorámica, podemos rotarla artificialmente simplemente desplazando las columnas de la misma un determinado número de posiciones. De esta manera, para cada una de las imágenes  $\vec{x}^j \in \mathfrak{R}^{M \times 1}; j=1 \dots N$  podemos construir una submatriz

$\mathbf{X}^j \in \mathfrak{R}^{M \times Q}$  donde la primera columna es la imagen original, y el resto de ellas son versiones rotadas de la primera, con una rotación de valor  $2\pi/Q$  entre ellas.

Quando tenemos un conjunto de  $N$  imágenes de entrenamiento, la matriz de datos está compuesta por  $N$  bloques, y la matriz de covarianza tiene la siguiente forma:

$$\mathbf{X} = [\mathbf{X}^1 | \mathbf{X}^2 | \dots | \mathbf{X}^N]$$

$$\Rightarrow C = X^T X = \begin{bmatrix} X^{11} & X^{12} & \dots & X^{1N} \\ X^{21} & X^{22} & \dots & X^{2N} \\ \vdots & \vdots & \ddots & \vdots \\ X^{N1} & X^{N2} & \dots & X^{NN} \end{bmatrix} \quad (1)$$

Donde  $\mathbf{X}^{ik} \in \mathfrak{R}^{Q \times Q}$  son bloques circulantes. Los vectores propios de una matriz general circular son los  $Q$  vectores base de la matriz de Fourier [10]:

$$\vec{\omega}_i = [1 \quad \gamma^i \quad \gamma^{2i} \quad \dots \quad \gamma^{(Q-1)i}]^T$$

$$i = 0, \dots, Q, \quad \gamma = e^{-2\pi j/Q}, \quad j = \sqrt{-1} \quad (2)$$

Esta propiedad nos permite calcular los vectores propios sin necesidad de realizar la descomposición SVD de  $C$  (esto sería un proceso muy costoso desde el punto de vista computacional). Este problema puede ser resuelto llevando a cabo  $Q$  descomposiciones de orden  $N$ . Los vectores propios de  $C$  serán de la forma:

$$\vec{v}_i = [\alpha_i^1 \vec{\omega}_i^T, \alpha_i^2 \vec{\omega}_i^T, \dots, \alpha_i^N \vec{\omega}_i^T]^T, i = 0, \dots, N \quad (3)$$

donde  $\vec{\alpha}_i = [\alpha_i^1, \alpha_i^2, \dots, \alpha_i^N]^T, i = 0, \dots, N$  son los vectores propios de la siguiente matriz:

$$\Lambda = \begin{bmatrix} \lambda_i^{11} & \lambda_i^{12} & \dots & \lambda_i^{1N} \\ \lambda_i^{21} & \lambda_i^{22} & \dots & \lambda_i^{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_i^{N1} & \lambda_i^{N2} & \dots & \lambda_i^{NN} \end{bmatrix} \quad (4)$$

donde  $\lambda_i^{jk}$  es el valor propio de  $X^{jk}$  correspondiente al vector propio  $\vec{\omega}_i$ . Como la matriz  $\Lambda$  tiene  $N$  vectores propios, si repetimos este proceso para cada  $\vec{\omega}_i$  podemos obtener  $Q \cdot N$  vectores propios de  $C$  linealmente independientes.

## 2.2 TÉCNICAS BASADAS EN LA TRANSFORMADA DE FOURIER

### 2.2.1 Transformada Discreta de Fourier 2D

Quando tenemos una imagen  $f(x,y)$  con  $N_y$  filas y  $N_x$  columnas, la Transformada Discreta de Fourier 2D se define mediante:

$$\begin{aligned}\mathfrak{F}[f(x, y)] &= F(u, v) = \\ &= \frac{1}{N_x N_y} \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} f(x, y) \cdot e^{-2\pi j \left( \frac{ux}{N_x} + \frac{vy}{N_y} \right)}\end{aligned}\quad (5)$$

$$u = 0, \dots, N_x - 1, v = 0, \dots, N_y - 1$$

Los componentes de la imagen transformada son números complejos, por lo que se puede separar en dos matrices, una con los módulos (espectro de potencia) y otra con las fases. La información más relevante en el dominio de Fourier se concentra en los componentes de baja frecuencia. Además, la eliminación de la información de alta frecuencia puede conducir a una mejora en la localización porque esas componentes son las que se encuentran más afectadas por posible ruido. Otra propiedad interesante cuando trabajamos con imágenes panorámicas es la invarianza en la rotación, tal y como expresa la propiedad de traslación:

$$\mathfrak{F}[f(x - x_0, y - y_0)] = F(u, v) \cdot e^{-2\pi j \left( \frac{ux_0}{N_x} + \frac{vy_0}{N_y} \right)}\quad (6)$$

$$u = 0, \dots, N_x - 1, v = 0, \dots, N_y - 1$$

De acuerdo con esta propiedad, el espectro de potencia de la imagen rotada sigue siendo el mismo que el de la imagen original y sólo se produce un cambio en la fase de los componentes de la imagen transformada, cuyo valor depende del desplazamiento en el eje  $x$  ( $x_0$ ) y en el eje  $y$  ( $y_0$ ). Esto significa que cuando se adquieren dos imágenes en puntos cercanos del entorno pero con diferentes orientaciones del robot, el espectro de potencia es muy similar en ambos casos. Estudiando las diferencias en las fases podríamos estimar el ángulo relativo entre las dos orientaciones, usando la ecuación (6).

### 2.2.2 Firma de Fourier de la imagen

Si trabajamos con imágenes panorámicas, podemos usar otra representación compacta basada en la transformada discreta de Fourier, que presenta la misma propiedad de traslación, cuando se aplica a imágenes panorámicas [6]. Consiste en expandir cada fila de la imagen panorámica  $\{a_n\} = \{a_0, a_1, \dots, a_{N_y-1}\}$ , usando la Transformada Discreta de Fourier, en la secuencia de números complejos  $\{A_n\} = \{A_0, A_1, \dots, A_{N_y-1}\}$ .

Esta firma de Fourier presenta las mismas propiedades que la Transformada de Fourier 2D. La información más importante queda concentrada en los componentes de baja frecuencia de cada fila, y presenta invarianza a la rotación. Asimismo, explota mejor dicha invarianza ante rotación en el plano en

que se mueve el robot. Estas rotaciones conducen a dos imágenes iguales pero desplazadas a lo largo del eje horizontal (fig. 1). Cada fila de la primera imagen puede ser representada con la secuencia  $\{a_n\}$  y cada fila de la segunda imagen será  $\{a_{n-q}\}$ , siendo  $q$  la cantidad de desplazamiento, que es proporcional a la rotación relativa entre imágenes. La invarianza rotacional es deducida del teorema del desplazamiento, que puede ser expresado como:

$$\mathfrak{F}\{\{a_{n-q}\}\} = A_k e^{-j \frac{2\pi qk}{N_y}}; \quad k = 0, \dots, N_y - 1 \quad (7)$$

donde  $\mathfrak{F}\{\{a_{n-q}\}\}$  es la Transformada de Fourier de la secuencia desplazada, y  $A_k$  son los componentes de la Transformada de Fourier de la secuencia no desplazada. De acuerdo con esta expresión, la amplitud de la Transformada de Fourier de la imagen desplazada es la misma que la de la transformada de la original y sólo hay un cambio de fase, proporcional a la cantidad de desplazamiento  $q$ .



Figura 1: Una rotación del robot en el plano de la superficie produce un desplazamiento en la imagen panorámica capturada

## 3 CONSTRUCCIÓN DEL MAPA

Para llevar a cabo el experimento, hemos capturado un conjunto de imágenes omnidireccionales sobre una rejilla predefinida de 20x20 cm en un entorno interior, incluyendo una zona no estructurada (un laboratorio) y una estructurada (un pasillo). Trabajamos con imágenes panorámicas con un tamaño de 56x256 píxeles. Una vez que tenemos todas las imágenes panorámicas, hemos utilizado los métodos de compresión expuestos en la sección previa. La fig. 2(a) muestra una vista en planta de la cuadrícula usada para tomar las imágenes y dos ejemplos de imágenes panorámicas.

Antes de usar los métodos de compresión, se ha llevado a cabo una normalización y un proceso de filtrado para hacer un mapa robusto frente a cambios en la iluminación de las escenas. Hemos observado como se producen diferencias significativas cuando

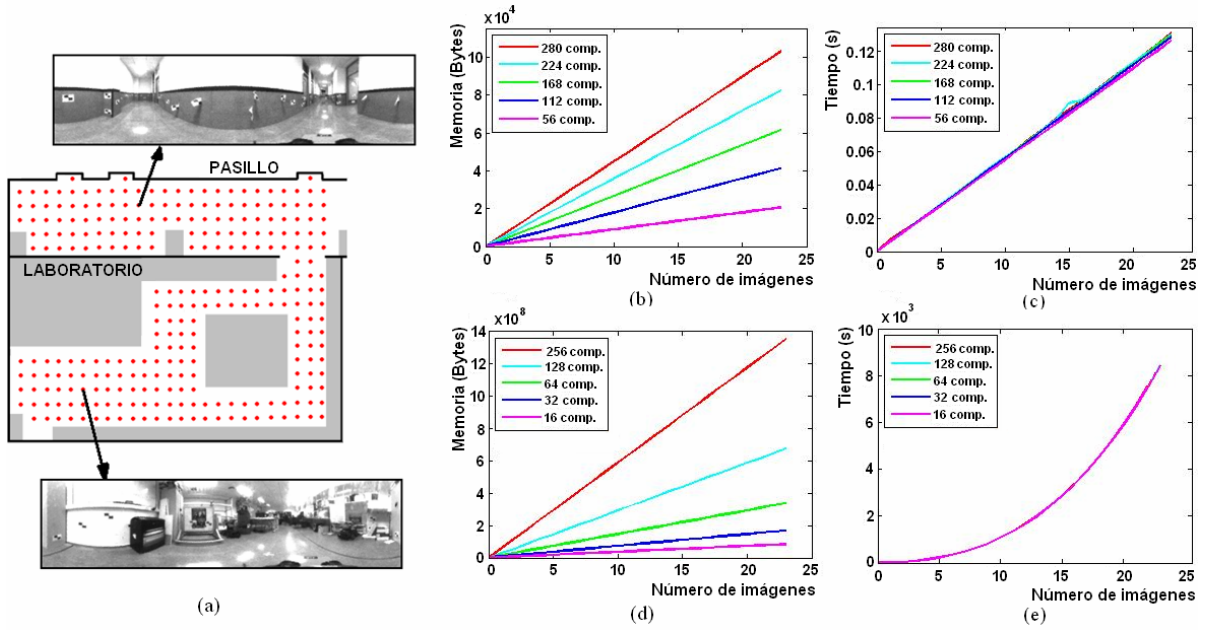


Figura 2: (a) Rejilla usada para capturar el conjunto de imágenes de entrenamiento, (b) cantidad de memoria utilizada y (c) tiempo necesario para la construcción del mapa con los enfoques basados en la Transformada de Fourier, (d) cantidad de memoria y (e) tiempo transcurrido con el enfoque basado en PCA

usamos cada uno de los métodos de compresión, con respecto al tiempo transcurrido y la cantidad de memoria que ocupa el mapa una vez construido.

La fig. 2(b) muestra la cantidad de memoria que ocupa el mapa, dependiendo del número de imágenes que contiene, y del número de componentes de Fourier que se retienen (los de menor frecuencia). Como ejemplo, cuando se trabaja con la firma de Fourier, tomar 224 componentes implica quedarse 4 componentes por fila (56 filas en la imagen). En el caso de la Transformada de Fourier 2D, 224 componentes significa que tomamos 7 filas y 32 columnas (donde se concentra la información principal).

La fig. 2(c) muestra el tiempo que transcurre durante la construcción de la base de datos, dependiendo del número de imágenes en el mapa y del número de componentes de Fourier usados. El tiempo transcurrido es muy similar cuando usamos la Firma de Fourier y la Transformada de Fourier 2D. Por otro lado, la fig. 2(d) y 2(e) muestran los resultados cuando usamos la técnica de compresión PCA para imágenes rotadas. La fig 2(d) muestra la cantidad de memoria requerida. El mapa PCA esta compuesto de la matriz  $V \in C^{K \times M}$ , que contiene los  $K$  principales vectores propios y las proyecciones de las imágenes de entrenamiento  $\vec{p}_j \in C^{K \times 1}$ ;  $j = 1 \dots N$ .

Aunque las imágenes de entrenamiento han sido rotadas artificialmente para añadir la información de orientación a la base de datos, no es necesario almacenar las proyecciones de todas las imágenes

rotadas, sino solamente las proyecciones de una orientación por cada imagen de entrenamiento. Esto es debido al hecho de que una rotación en la imagen equivale a un cambio en el ángulo de los coeficientes PCA de esta, pero no en el módulo. Por tanto, si tenemos los coeficientes para un punto de vista representativo, los coeficientes de las imágenes rotadas pueden ser generados mediante una rotación en el plano complejo. De este modo, el módulo de las proyecciones puede usarse para calcular la posición del robot, y la fase es útil para conocer la orientación. De todos modos, como podemos ver en la fig. 2, PCA es un proceso apreciablemente más costoso computacionalmente comparado con la Transformada de Fourier.

#### 4 LOCALIZACIÓN Y CÁLCULO DE LA ORIENTACIÓN DEL ROBOT

Para probar la validez de los mapas construidos, se han capturado con el robot varias imágenes de test en algunos puntos intermedios del entorno. Hemos capturado dos conjuntos de imágenes de test, la primera, al mismo tiempo que tomábamos el conjunto de prueba y la segunda unos días después, en diferentes momentos del día (con distintas condiciones de iluminación) y cambiando la posición de algunos objetos. El objetivo es el cálculo de la posición y orientación del robot cuando se toman imágenes de prueba, solamente usando la información visual del mapa.

#### 4.1 TÉCNICAS BASADAS EN PCA

El mapa PCA se compone de la matriz  $V \in C^{K \times M}$ , que contiene los  $K$  vectores propios principales y las proyecciones de las imágenes de entrenamiento  $\vec{p}^j \in C^{K \times 1}$  (una por posición, como se explicó en la sección anterior), que han sido divididas en dos vectores,  $\vec{p}_m^j \in R^{K \times 1}$  (módulo de las componentes de la proyecciones) y  $\vec{p}_{ph}^j \in R^{K \times 1}$  (fases).

Para calcular la ubicación en la que el robot tomó cada imagen de prueba, tenemos que proyectar la imagen de prueba  $\vec{x}^i \in R^{M \times 1}$  en el espacio propio,  $\vec{p}^i = V^T \cdot \vec{x}^i \in C^{K \times 1}$ . Entonces, calculamos el vector de módulos  $\vec{p}_m^i \in R^{K \times 1}$  y lo comparamos con todos los vectores  $\vec{p}_m^j$  guardados en el mapa, usando como criterio la distancia Euclídea. La posición correspondiente al robot es extraída como la que presenta una menor distancia. Una vez conocemos la posición, usamos el vector de fases  $\vec{p}_{ph}^i$  para calcular la orientación del robot.

La tabla 1 muestra los resultados que obtenemos cuando calculamos la posición y la orientación cuando el conjunto de entrenamiento se ha tomado sobre una cuadrícula de 20x20 cm y la tabla 2 muestra los mismos resultados para una cuadrícula de 40x40 cm, dependiendo del número de vectores propios ( $K$ ) que se tome. En estas tablas,  $p_1$  es la probabilidad de que la imagen correcta sea la calculada por el método (la que ha suministrado la menor distancia),  $p_2$  es la probabilidad de que la imagen correcta esté entre las dos primeras que devuelve el método, y  $p_3$  es la probabilidad de que sea una de las tres imágenes devueltas. Finalmente,  $e_\theta$  es el error medio en la estimación de la orientación.

Tabla 1: Precisión en la estimación de la posición y orientación con el método PCA. Conjunto de entrenamiento sobre rejilla 20x20 cm.

$K$	$p_1$	$p_2$	$p_3$	$e_\theta$
32	77.8%	81.3%	88.6%	10.0°
64	82.1%	83.1%	84.2%	9.1°
128	82.1%	84.8%	85.4%	7.9°

Tabla 2: Precisión en la estimación de la posición y orientación con el método PCA. Conjunto de prueba sobre rejilla 40x40 cm.

$K$	$p_1$	$p_2$	$p_3$	$e_\theta$
32	67.1%	70.9%	72.2%	21.6°
64	68.5%	72.3%	74.9%	18.2°
128	65.4%	67.3%	71.3%	23.4°

La fig. 3 muestra el tiempo tomado por este método para calcular la posición y orientación, dependiendo del número de imágenes almacenadas en el mapa, y del número de vectores propios.

#### 4.2 TÉCNICAS BASADAS EN LA TRANSFORMADA DE FOURIER

Para calcular la posición y orientación del robot para cada imagen de prueba, calculamos la Transformada de Fourier (con los dos métodos descritos en la sección previa) y entonces, calculamos la distancia Euclídea del espectro de potencia de la imagen de prueba con respecto al espectro almacenado en el mapa. La que mejor corresponda es tomada como la posición actual del robot.

Por otro lado, la orientación es calculada con la ecuación (6), cuando trabajamos con la Transformada de Fourier Discreta 2D (asumiendo  $y_0 = 0$ ) y con la expresión (7), cuando trabajamos con la firma de Fourier. En este caso se obtiene un ángulo diferente para cada fila por lo que obtenemos el ángulo medio. Las tablas 3 y 4 muestran la precisión que obtenemos en la estimación de la posición y la orientación con los métodos de Fourier.

Tabla 3: Precisión en la estimación de la posición y orientación. Métodos de Fourier. Conjunto de entrenamiento sobre rejilla 20x20 cm.

$K$	$p_1$	$p_2$	$p_3$	$e_\theta$
16x56	91.5%	93.2%	94.6%	7.0°
64x56	93.6%	96.8%	97.0%	6.8°
125x56	92.0%	94.1%	96.0%	6.1°

Tabla 4: Precisión en la estimación de la posición y orientación. Métodos de Fourier. Conjunto de entrenamiento sobre rejilla 40x40 cm.

$K$	$p_1$	$p_2$	$p_3$	$e_\theta$
16x56	83.3%	87.7%	89.1%	14.2°
64x56	85.2%	89.2%	91.4%	13.9°
125x56	86.1%	89.3%	90.0%	13.1°

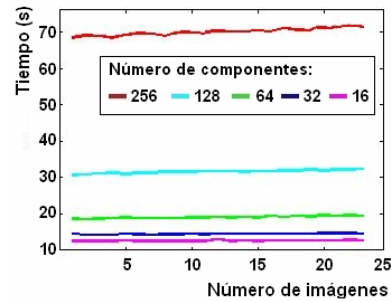


Figura 3: Tiempo consumido en calcular la posición y orientación con PCA

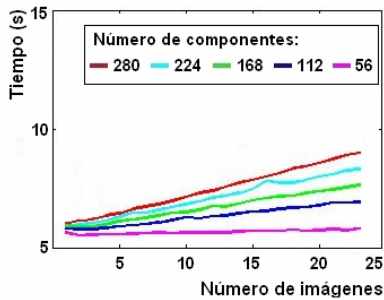


Figura 4: Tiempo consumido en calcular la posición y orientación con los métodos basados en la Transformada de Fourier.

La fig. 4 muestra el tiempo necesario desde que el robot captura la imagen hasta que obtiene la posición y orientación del robot, dependiendo del número de imágenes almacenadas en el mapa y del número de componentes de Fourier que nos quedemos. Este método mejora claramente los métodos basados en PCA tanto en precisión como en tiempo consumido.

## 5 CONCLUSIONES

En este trabajo, hemos expuesto los principios de la creación de un mapa denso de un entorno real, usando imágenes omnidireccionales y métodos basados en apariencia. Hemos presentado tres métodos diferentes para comprimir la información en el mapa. Las propiedades matemáticas de estos métodos junto con la rica información suministrada por las imágenes omnidireccionales obtenidas del entorno permite al robot calcular su posición y orientación dentro del mapa.

El método de la Transformada de Fourier (la Transformada de Fourier Discreta 2D y la Firma de Fourier) ha demostrado ser un buen método para comprimir la información comparado con PCA respecto al tiempo y la cantidad de memoria, y la precisión en la estimación de la posición y de la orientación. Otra propiedad importante es que la Transformada de Fourier es un método inherentemente incremental. Cuando trabajamos con PCA, es necesario disponer de todas las imágenes de entrenamiento disponibles antes de llevar a cabo la compresión por lo que este método no se puede aplicar a tareas que requieran un proceso incremental (ej. un algoritmo SLAM donde la información de la nueva localización debe ser añadida al mapa mientras el robot se mueve alrededor del entorno). La transformada de Fourier no presenta esta desventaja porque la compresión de cada imagen es obtenida independientemente. Estas propiedades podrán aplicarse en futuras tareas donde los robots tengan

que añadir la nueva información al mapa y localizarse asimismo en tiempo real.

Ese trabajo abre la puerta a nuevas aplicaciones de los métodos basados en apariencia en robots móviles. Como hemos mostrado, el principal problema que presentan estos métodos es el alto requerimiento de memoria y tiempo de cálculo para construir la base de datos y hacer las necesarias comparaciones para calcular la posición y orientación del robot. Una vez hemos estudiado en profundidad algunos métodos de compresión de información y separado el cálculo de la posición y la orientación, el siguiente paso debe ser poner a prueba la robustez frente a cambios en la iluminación y en la posición de algunos objetos en la escena. Además, su robustez y simplicidad los hacen aplicables a la creación de mapas más sofisticados, donde no tenemos información sobre la posición del robot cuando tomó las imágenes de entrenamiento.

## Agradecimientos

Este trabajo está apoyado por el gobierno español a través del proyecto DPI2007-61197. 'Sistemas de percepción visual móvil y cooperativo como soporte para la realización de tareas con redes de robots'.

## Referencias

- [1] Artac, M.; Jogan, M. & Leonardis, A., 2002. Mobile Robot Localization Using an Incremental Eigenspace Model, In *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, USA, pp. 1205-1030, IEEE.
- [2] Booij, O., Terwijn, B., Zivkovic, Z., Kröse, B., 2007. Navigation using an Appearance Based Topological Map. In *IEEE International Conference on Robotics and Automation*, pp. 3297-3932 IEEE Press, New York.
- [3] Jogan, M., Leonardis, A., 2000. Robust Localization Using Eigenspace of Spinning-Images. In *Proc. IEEE Workshop on Omnidirectional Vision*, Hilton Head Island, USA, pp. 37-44, IEEE.
- [4] Kirby, M., 2000. *Geometric data analysis. An empirical approach to dimensionality reduction and the study of patterns*, Wiley Interscience.
- [5] Kröse, B., Bunschoten, R., Hagen, S., Terwijn, B. Vlassis, N., 2004. Household robots: Look and learn. In *IEEE Robotics & Automation magazine*. Vol. 11, No. 4, pp. 45-52.
- [6] Menegatti, E.; Maeda, T. Ishiguro, H., 2004. Image-based memory for robot navigation using

properties of omnidirectional images. In *Robotics and Autonomous Systems*. Vol. 47, No. 4, pp. 251-276.

- [7] Payá, L., Reinoso, O., Gil, A., Sogorb, J., 2008. Multi-robot route following using omnidirectional vision and appearance-based representation of the environment. In *Lecture Notes in Artificial Intelligence. Hybrid Artificial Intelligence Systems*, Vol. 5271, pp. 680-687 Springer.
- [8] Rossi, F., Ranganathan, A., Dellaert, F., Menegatti, E., 2008. Toward topological localization with spherical Fourier transform and uncalibrated camera. In *Proc. Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots*. Venice (Italy), pp. 319-330.
- [9] Thrun, S., 2003. Robotic Mapping: A Survey, In *Exploring Artificial Intelligence in the New Milenium*, pp. 1-35, Morgan Kaufmann Publishers, San Francisco, USA.
- [10] Ueonara, M., Kanade, T, 1998. Optimal approximation of uniformly rotated images: relationship between Karhunen-Loeve expansion and Discrete Cosine Transform. In *IEEE Transactions on Image Processing*. Vol. 7, No. 1, pp. 116-119.
- [11] Vasudevan, S., Gächter, S., Nguyen, V., Siegwart, R., 2007. Cognitive maps for mobile robots – an object based approach. In *Robotics and Autonomous Systems*. Vol. 55, No. 1, pp. 359-371.