

# LOCALIZACIÓN DE ROBOTS MÓVILES CON 4 GDL MEDIANTE VISIÓN OMNIDIRECCIONAL

C. Parra, L.M. Jiménez, M. Ballesta, O. Reinoso y D. Valiente

Departamento de Ingeniería de Sistemas y Automática.

Universidad Miguel Hernández de Elche. Avda. de la Universidad s/n. 03202, Elche (Alicante), Spain  
{cristobal.parra01, luis.jimenez, m.ballesta, o.reinoso, dvaliente}@umh.es

## Resumen

*La localización de robots, junto a la creación de mapas y planificación, es una tarea fundamental que permite la navegación autónoma de estos dispositivos en un entorno desconocido. En este trabajo se presenta un método para calcular la transformación entre dos poses del robot que se desplaza en un entorno en el que pueden darse cambios en la altura. Estos robots están provistos de un sistema omnidireccional que les permite obtener puntos característicos del entorno. A partir de estos puntos, aplicamos el método que se describe en este trabajo, para obtener la localización del robot salvo un factor de escala sin tener conocimiento previo de la escena.*

**Palabras clave:** Robótica móvil, visión omnidireccional, localización.

## 1. INTRODUCCIÓN

En el campo de la navegación autónoma de robots, los sensores visuales adquieren gran relevancia dado su bajo coste y la capacidad de extraer mayor cantidad de información del entorno [5]. Los sensores catadióptricos destacan en este ámbito. Estos sistemas físicos están provistos de una cámara y un espejo superando a otros dispositivos al poder adquirir una visión global del entorno con una única imagen [12].

En este trabajo, los robots están provistos de una cámara omnidireccional con la que se obtienen vistas del entorno [9]. Se denomina vista a una imagen omnidireccional que contiene la escena de 360 grados en torno al robot. A partir de estas vistas, se obtienen puntos característicos o marcas visuales [7]. Las marcas visuales se definen como puntos detectados en cada vista que permiten establecer correspondencias entre dos vistas consecutivas distintas. Estas marcas se corresponden con puntos físicos del entorno del robot, como puede ser la esquina de un objeto [4]. A partir de estas marcas, se pretende que el robot sea capaz de localizarse y construir un mapa del entorno.

El objetivo principal de este trabajo es el desarro-

llo de un método que permita obtener la transformación entre dos poses consecutivas del robot a partir de las marcas obtenidas entre dos vistas capturadas por una cámara omnidireccional. Este trabajo es la continuación de una contribución previa presentada por [2] en la que se obtiene el desplazamiento de un robot a lo largo de un mismo plano. En el presente trabajo, no se restringe el movimiento del robot a un solo plano, sino que el método desarrollado también es válido para cambios en altura. Concretamente, estableciendo correspondencias entre dos vistas, se obtiene la traslación salvo un factor de escala, rotación y el cambio en altura entre dos poses consecutivas.

De las vistas omnidireccionales se obtienen correspondencias que relacionan dos poses consecutivas del robot. A partir de estas marcas o correspondencias, y con el método desarrollado, basado en el cálculo de la matriz esencial y teniendo en cuenta cambios en altura, se obtiene la transformación entre dichas poses del robot salvo un factor de escala. Se trata de un paso necesario para poder calcular la odometría visual del robot y de este modo, poder obtener la localización del robot en un entorno desconocido con posibles cambios en altura a partir, únicamente, de información visual.

El resto del artículo se estructura como se describe a continuación. En la Sección 2 se detalla el sistema de adquisición omnidireccional así como el modelo de movimiento que define el avance del robot en el entorno. En la Sección 3 se describe el método desarrollado para estimar la transformación entre poses a partir de imágenes omnidireccionales. Esta sección incluye el cálculo de la matriz esencial que relaciona ambas vistas así como el sistema de ecuaciones que nos permite calcular la transformación entre las poses del robot. En la Sección 4, se presentan las pruebas realizadas y los resultados que validan el método presentado. Por último, en la Sección 5 se incluyen las principales conclusiones y líneas futuras.

## 2. SISTEMA DE ADQUISICIÓN DE IMÁGENES OMNIDIRECCIONALES

Consideramos el caso en el que el robot está equipado con una única cámara omnidireccional, según se muestra en la Figura 1. Cuando el robot se mueve por el entorno captura imágenes omnidireccionales de las que se pueden extraer un conjunto de puntos de interés. A continuación, se buscan correspondencias con la imagen omnidireccional correspondiente a la posición anterior en el mapa. Si se encuentra un número suficiente de correspondencias entre las imágenes, entonces se puede calcular la rotación y traslación (salvo un factor de escala) entre ambas capturas. A partir de estas medidas, es posible deducir la localización del robot con respecto a las poses anteriores en el mapa. El cálculo de la rotación y traslación está detallado en la Sección 3.

Los ángulos relativos que relacionan el movimiento entre dos imágenes omnidireccionales son  $\alpha$ ,  $\phi$  y  $\beta$ , que pueden verse en las Figuras 1 y 2. Estos ángulos representan la posición relativa del robot y permiten su localización. Para su obtención deben detectarse puntos característicos en ambas imágenes y encontrar sus correspondencias.

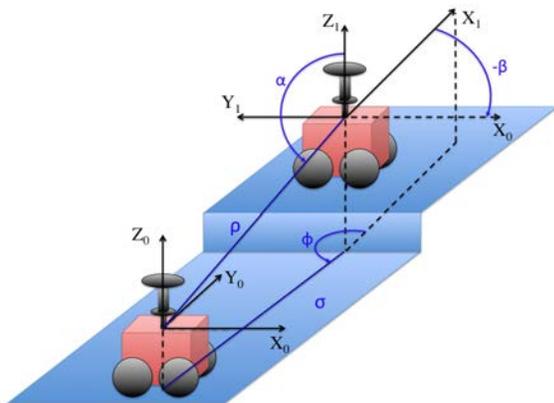


Figura 1: Representación tridimensional de los sistemas de coordenadas entre dos poses del robot.

### 2.1. DETECCIÓN DE PUNTOS CARACTERÍSTICOS Y CORRESPONDENCIAS

La técnica que se plantea a continuación, utiliza información de la dirección y magnitud del desplazamiento del robot mediante únicamente información visual. Con el propósito de calibrar el sistema visual y obtener las distintas proyecciones de las escenas, se utiliza la librería de Matlab OCamCalib [11], que de entre sus diferentes opciones, dado

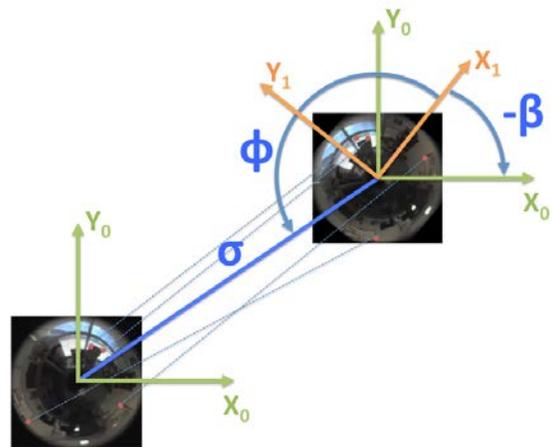


Figura 2: Dos imágenes omnidireccionales con correspondencias entre ellas, junto con la proyección sobre el plano de los dos sistemas de coordenadas.

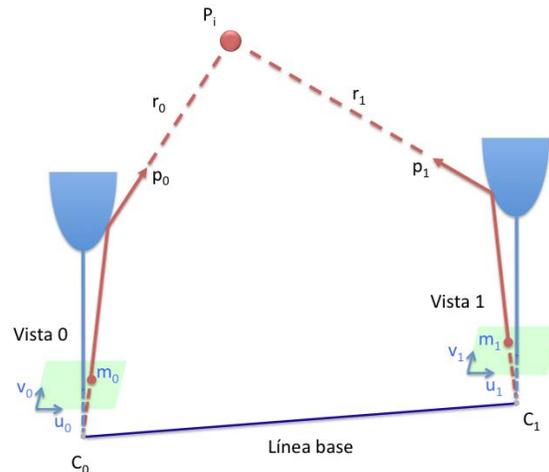


Figura 3: Esquema de generación de un punto característico en dos vistas omnidireccionales distintas.

un punto  $m_i = [u_i, v_i, 1]^T$  de la vista  $i$  capturada en la posición  $K_i$ , se puede obtener un vector  $p_i = [w_{x,i}, w_{y,i}, w_{z,i}]^T$  apuntando en la dirección del punto 3D  $P_i = [X_i, Y_i, Z_i]^T$  como se muestra en la Figura 3, de modo que

$$p_i = \begin{bmatrix} w_{x,i} \\ w_{y,i} \\ w_{z,i} \end{bmatrix} = \rho \begin{bmatrix} u_i \\ v_i \\ f(u_i, v_i) \end{bmatrix}, \quad (1)$$

donde  $f(u_i, v_i)$  es una función polinomial cuyos coeficientes se obtienen con la calibración.

Se emplea el algoritmo SURF (Speeded Up Robust Features) [1] con el fin de obtener puntos de interés entre imágenes. Según [4], SURF mejora los resultados de otros detectores y descriptores

en términos de robustez de los puntos detectados y de invarianza del descriptor. En la Figura 2 se presentan dos imágenes omnidireccionales donde se han indicado un conjunto de correspondencias.

## 2.2. MODELO DE MOVIMIENTO

Se asumirá que el robot se encuentra en una posición  $K_i$  de la que ha capturado una imagen y se requiere conocer la posición actual respecto a la pose anterior  $K_{i-1}$ . Los ángulos  $(\phi, \alpha, \beta)$  que relacionan la nueva pose con respecto a la anterior se representan en la Figura 1. El ángulo  $\phi$  es la orientación con la que la vista  $i$  es observada desde el sistema de referencia móvil asociado al robot,  $\beta$  es la orientación relativa entre ambas imágenes y  $\alpha$  es el desplazamiento en altura de la nueva pose del robot con respecto a la anterior.

El cálculo de estos tres ángulos nos permite conocer la transformación entre dos poses consecutivas del robot.

## 3. ESTIMACIÓN DE LA POSICIÓN DEL ROBOT A PARTIR DE IMÁGENES OMNIDIRECCIONALES

En este trabajo, se extraen características visuales de la imagen omnidireccional. Estas marcas visuales se hacen corresponder entre dos vistas consecutivas del robot.

### 3.1. MATRIZ ESENCIAL

Una vez detectados los puntos SURF y determinadas sus correspondencias entre imágenes, ha de establecerse un proceso para recuperar los ángulos que definen el movimiento del robot.

La condición de epipolaridad establece la relación entre dos puntos 3D observados desde diferentes vistas

$$p_1^T E p_0 = 0. \quad (2)$$

La matriz  $E$  representa la matriz esencial y puede ser calculada a partir de una serie de puntos correspondientes entre imágenes. Un punto  $p_i = [w_{x,i}, w_{y,i}, w_{z,i}]^T$  define las coordenadas normalizadas de un punto respecto a la cámara  $i$ . En nuestro caso, debido al uso de una única cámara, las imágenes se adquieren desde dos posiciones indeterminadas, sin información de la distancia entre ellas. Este hecho implica el desconocimiento de la profundidad y por tanto la solución se obtiene salvo un factor de escala  $\rho$ .

Si  $I_0$  es la imagen capturada en la pose anterior, y la imagen  $I_1$  corresponde a la pose actual, los centros ópticos para las imágenes adquiridas se definen como  $C_0$  y  $C_1$ .

Se puede definir la línea base  $\mathbf{b}_1 = [b_1^x, b_1^y, b_1^z]^T$  como la dirección de la traslación de la pose  $K_1$  del robot con respecto a la pose  $K_0$  y  $\mathbf{R}_1$  la orientación entre las coordenadas de referencia. Se define la línea base actual como  $t_1 = b_1 \mathbf{b}_1$ , siendo  $b_1$  la norma de la línea base.

De este modo, se pueden relacionar las coordenadas de los puntos característicos vistos desde cada pose del robot según

$$P_0 = t_1 + \mathbf{R}_1 P_1. \quad (3)$$

Así, si se define  $P_i = r_i p_i$  donde  $r_i$  es un escalar que codifica la información de profundidad de la escena, la ecuación anterior queda expresada como

$$r_0 p_0 = b_1 \mathbf{b}_1 + r_1 \mathbf{R}_1 p_1, \quad (4)$$

donde  $r_0$  y  $r_1$  son las profundidades y  $b_1$  es la longitud de la línea base, todos ellos parámetros desconocidos y  $p_i$  la dirección conocida del punto característico visto desde la cámara  $i$  (obtenido a través de la calibración de la cámara a partir de la vista  $i$  como se explica en la Sección 2.1). Por tanto, al no disponer de información acerca de estos parámetros, la reconstrucción de la escena puede ser llevada a cabo salvo un factor de escala.

Para estimar el movimiento del robot, es necesario obtener un mínimo de correspondencias entre puntos característicos de cada imagen según se explica en la sección anterior. Los puntos característicos de la imagen capturada en la primera posición, pueden ser relacionados con los de la segunda a partir de la geometría epipolar. La geometría epipolar expresa el hecho de que para un rayo  $r_0$  originado en  $C_0$ , la primera cámara, y un rayo  $r_1$  originado en  $C_1$ , la segunda, estos rayos coinciden en un único punto del espacio tridimensional y se cumple que

$$p_0 \cdot (\mathbf{b}_1 \times (\mathbf{R}_1 p_1)) = p_0^T S R p_1 = p_0^T E p_1 = 0, \quad (5)$$

donde  $E$  es la matriz esencial definida previamente y  $S$  es la matriz antisimétrica del producto vectorial.

La matriz esencial puede ser estimada si se conocen las correspondencias de al menos 8 puntos característicos de cada imagen, de modo que cada correspondencia produce una restricción. Esta matriz no es única, sino que existen infinitas soluciones para la matriz esencial que cumplen con las condiciones necesarias. Existen diferentes métodos como el propuesto en [6] para el cálculo de la matriz esencial y poder obtener el movimiento del

robot, de manera que se obtienen un conjunto de posibles soluciones de las cuales, tan solo una de ellas es posible. En la Sección 3.5, se describe el método a partir del cual obtener la solución correcta de entre las diferentes soluciones. En este trabajo, se aporta un nuevo enfoque en el que debido a que se estudia un movimiento con 4 GDL, se pueden incluir nuevas restricciones (definidas en la Sección 3.3) que permiten obtener una matriz más robusta, donde los coeficientes de la matriz se conservan independientemente de los puntos con los que ha sido obtenida, y siendo el mínimo número de puntos necesarios  $n = 4$  (con la particularidad de disponer de una buena semilla inicial como se explica en la Sección 3.3).

### 3.2. RESTRICCIONES EN EL MOVIMIENTO

Según el nuevo enfoque de este trabajo, el movimiento de la cámara está restringido a un único giro alrededor del eje  $Z$  y al movimiento a lo largo de los ejes  $X, Y, Z$ .

De este modo, la línea base que une la posición del robot entre dos posiciones distintas, puede definirse de la siguiente manera

$$\mathbf{b}_1 = [\cos(\phi)\sin(\alpha), \sin(\phi)\sin(\alpha), \cos(\alpha)] \quad (6)$$

y la rotación de un ángulo  $\beta$  en torno al eje  $Z$ , de modo que la matriz esencial queda definida de la siguiente forma

$$E = SR, \quad (7)$$

donde

$$S = \begin{bmatrix} 0 & -\cos(\alpha) & \sin(\phi)\sin(\alpha) \\ \cos(\alpha) & 0 & -\cos(\phi)\sin(\alpha) \\ -\sin(\phi)\sin(\alpha) & \cos(\phi)\sin(\alpha) & 0 \end{bmatrix} \quad (8)$$

y dado que únicamente se considera una rotación alrededor del eje  $Z$

$$R = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

obteniendo como resultado la Ecuación (13).

### 3.3. ESTIMACIÓN DE LA MATRIZ ESENCIAL

Gracias a que la matriz esencial  $E$  representa una rotación  $R$  y una traslación  $T$  (salvo un factor de escala) entre los sistemas de referencia de dos

imágenes, a partir de la Ecuación (7) los ángulos que relacionan el movimiento del robot pueden ser obtenidos de la descomposición de  $E$ . Si reescribimos la matriz esencial como una matriz esencial generalista, obtendremos

$$E = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & 0 \end{bmatrix} \quad (10)$$

La matriz esencial puede ser estimada a partir de un conjunto de  $n$  correspondencias de dos vistas  $\{(p_{0,1}, p_{1,1}), \dots, (p_{0,n}, p_{1,n})\}$ , de modo que se cumplan las condiciones de la Ecuación (5). Escribiendo los valores distintos de cero de la matriz esencial generalista como un vector con la forma  $e = [e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8]^T$ , las  $n$  restricciones obtenidas a partir de las  $n$  correspondencias, pueden ser expresadas linealmente como  $De = 0$ , donde  $D$  tiene dimensión  $(n \times 8)$ . Cada fila  $D_j$  de  $D$  con  $j = (1, 2, \dots, n)$  tendrá la forma:

$$D_j = [x_0x_1, x_0y_1, x_0z_1, y_0x_1, y_0y_1, y_0z_1, z_0x_1, z_0y_1] \quad (11)$$

A estas restricciones, es necesario añadir también

$$\begin{aligned} \det(E) &= 0 \\ \|E\| &= 1 & e_1 &= e_5 \\ e_3e_2 + e_5e_6 &= \frac{e_7}{e_8} & -e_2 &= e_4 \end{aligned} \quad (12)$$

Una vez planteadas todas las restricciones necesarias, se resuelve el sistema mediante el método de Levenberg-Marquardt [8] para ecuaciones no lineales. Cabe destacar que para conseguir que la solución del sistema converja es importante partir de una buena semilla. Por ello, el punto inicial debe ser próximo a la solución final. En este caso se plantea la expresión  $De = 0$  y se busca el punto inicial utilizando una descomposición de valores singulares (SVD) [10] basada en el algoritmo de 8 puntos.

### 3.4. RECUPERACIÓN DEL MOVIMIENTO DEL ROBOT

Debido a las restricciones impuestas en el cálculo de  $E$ , esta matriz se encuentra normalizada y podremos recuperar los valores de los ángulos que definen el movimiento del robot salvo un factor de escala.

$$E = SR = \begin{bmatrix} -\cos(\alpha)\sin(\beta) & -\cos(\alpha)\cos(\beta) & \sin(\phi)\sin(\alpha) \\ \cos(\alpha)\cos(\beta) & -\cos(\alpha)\sin(\beta) & -\cos(\phi)\sin(\alpha) \\ \sin(\beta - \phi)\sin(\alpha) & \cos(\beta - \phi)\sin(\alpha) & 0 \end{bmatrix} \quad (13)$$

A continuación, obtendremos las posibles soluciones de  $\mathbf{R}_1$  y  $\mathbf{b}_1$ .

La línea base depende de  $\phi$  y  $\alpha$ , por lo que necesitamos obtener estos valores en primer lugar. El valor del ángulo  $\phi$  se calcula

$$\phi = \arctan\left(\frac{e_3}{-e_6}\right). \quad (14)$$

Sin embargo, nótese que debido a las singularidades de la función  $\arctan$ , el cálculo del ángulo  $\alpha$  supone evaluar cada uno de los valores de  $E$  para no caer en una indeterminación. De este modo, el valor de  $\alpha$  es  $\frac{\pi}{2}$  cuando  $e_1 = e_4 = 0$  y es 0 cuando  $e_3 = e_6 = e_7 = e_8 = 0$ , en otro caso debe emplearse la ecuación siguiente

$$\alpha = \arctan\frac{e_x}{e_y}, \quad (15)$$

donde  $\{e_x, e_y\}$  son los valores de las 8 posibles combinaciones restantes en cada caso.

Las cuatro posibles soluciones para la línea base por tanto se obtienen de la Ecuación (6) donde  $\phi$  toma el valor de la Ecuación (14) o  $\phi + \pi$  y  $\alpha$  toma el valor de (15) o  $\alpha + \frac{\pi}{2}$ .

El ángulo de rotación  $\beta$  puede obtenerse como

$$\beta = \arctan\left(\frac{e_7}{e_8}\right) + \arctan\left(\frac{e_3}{-e_6}\right), \quad (16)$$

y la rotación asociada a la línea base se obtiene de la Ecuación (9) con  $\beta$  el valor obtenido en la Ecuación (16) o la matriz correspondiente a un valor con ángulo  $\beta + \pi$ .

### 3.5. SELECCIÓN DE ROTACIÓN Y TRASLACIÓN

Para llevar a cabo la selección de la solución correcta, se puede utilizar la representación tridimensional de cada punto en el espacio y analizar la intersección de los rayos correspondientes al igual que en [2]. Evaluando todas las posibles combinaciones de  $R$  y  $b$ , se obtendrá un valor de  $r_1$  y  $r_0$  (la proyección de los rayos generados por cada marca visual), del que tomaremos como puntos válidos aquellos que devuelvan  $r_1 > 0$  y  $r_0 > 0$ . En cualquier otro caso, tomaremos el resultado como negativo y la solución no será tenida en cuenta.

Finalmente, la solución correcta será aquella que obtenga un mayor número de puntos válidos. Cabe destacar, que en ocasiones pueden producirse singularidades que produzcan que un punto sea negativo cuando debería ser positivo. Esto puede ocurrir en ciertos puntos, como los que se sitúan sobre la línea base.

## 4. EXPERIMENTOS Y RESULTADOS

En esta sección, se muestran los resultados obtenidos en simulación que permiten validar el método presentado anteriormente.

El experimento trata de obtener una estimación de la transformación entre cada pose. La simulación se ha realizado para diferentes posiciones del robot y con características visuales diferentes. El sistema de coordenadas global es el mostrado en la Figura 4. El eje de coordenadas del robot en  $K_0$  coincide con el sistema de coordenadas global. Una vez comenzada la trayectoria, el eje  $X$  coincide con la dirección del movimiento y al capturar una vista, el eje  $Z$  siempre permanece perpendicular al plano del suelo, es decir, el eje  $Z$  en el sistema de coordenadas del robot en todas y cada una de las poses coincide con el eje  $Z$  del sistema de coordenadas global, dado que la rotación se compone únicamente de una rotación en torno al eje  $Z$ .

El eje óptico de la cámara se supone instalado de manera perpendicular al plano del suelo como se aprecia en la Figura 2. Como consecuencia, una rotación del robot se corresponde con una rotación de la imagen respecto a su punto central.

Un conjunto de marcas han sido tomadas aleatoriamente a lo largo de la trayectoria seguida por el robot. Para cada transformación se han generado de manera aleatoria puntos del entorno con la intención de simular los de un caso real.

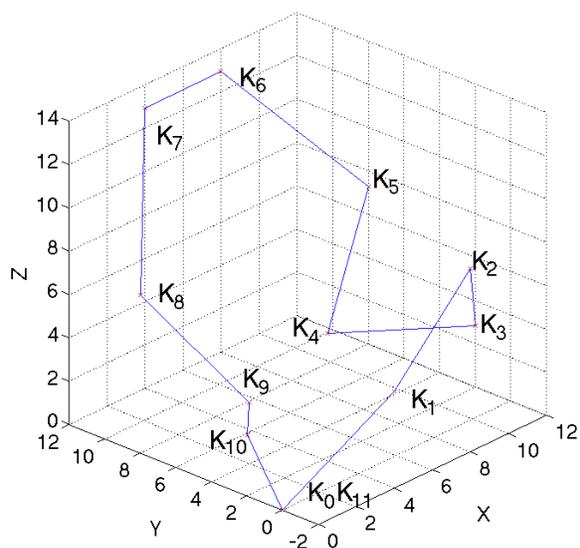


Figura 4: Trayectoria simulada llevada a cabo por el robot.

#### 4.1. Validación del algoritmo

Como caso previo se ha desarrollado una simulación con datos exactos para validar el algoritmo de localización.

La Figura 4 muestra la trayectoria seguida por el robot que se encuentra situado inicialmente en la pose  $K_0$  y tras capturar una imagen omnidireccional se desplaza a la pose siguiente  $K_1$ , donde captura una nueva imagen. En ese momento, se extraen puntos característicos de ambas vistas y se buscan sus correspondencias, para poder aplicar así el algoritmo propuesto en la Sección 3 y obtener la posición del robot en la pose  $K_1$  respecto a la pose anterior, en este caso  $K_0$ . Este proceso se desarrolla hasta llegar a  $K_{11}$ , última posición de la trayectoria, de manera que se cierra el bucle.

En la tabla 1 se puede apreciar como se obtienen resultados satisfactorios en la simulación coincidiendo con los ángulos esperados.

Tabla 1: Resultados de la validación del método de localización con la trayectoria simulada.

Poses	Ángulo $\beta$	Ángulo $\phi$	Ángulo $\alpha$
$K_0, K_1$	26.5651	180.0000	-225.0000
$K_1, K_2$	-26.5651	0.0000	-135.0000
$K_2, K_3$	-45.0000	-180.0000	45.0000
$K_3, K_4$	-153.4351	180.0000	90.0000
$K_4, K_5$	108.4349	180.0000	-225.0000
$K_5, K_6$	251.5651	360.0000	-135.0000
$K_6, K_7$	-45.0000	-180.0000	90.0000
$K_7, K_8$	-45.0000	180.0000	45.0000
$K_8, K_9$	-71.5651	180.0000	45.0000
$K_9, K_{10}$	116.5651	180.0000	90.0000
$K_{10}, K_{11}$	-90.0000	180.0000	54.6097

#### 4.2. Resultados en simulación con ruido añadido

Tras la validación anterior, se ha llevado a cabo un experimento en el que también se ha empleado la trayectoria mostrada en la Figura 4, pero en esta ocasión se ha añadido ruido gaussiano a las marcas visuales detectadas en la vista correspondiente a la pose  $K_i$  de manera que las correspondencias no fueran exactas.

De este modo, se puede comprobar la robustez del método ante posibles errores en la correspondencia entre los puntos detectados y la correspondencia llevada a cabo entre ellos. Se ha empleado un valor de  $\sigma = 5$  píxeles que afecta a la imagen tal y como se representa en la Figura 5, por lo que el error se añade directamente a los puntos detectados en la imagen. Esto es, dado que las marcas visuales

son simuladas y por tanto su correspondencia es exacta, una vez detectada la marca en la imagen omnidireccional de la pose  $K_i$  se le añade ruido de manera que se introduce cierta incertidumbre, pudiendo comprobar así el impacto de este tipo de errores en el algoritmo.

Tabla 2: Resultados del experimento de localización del robot incluyendo ruido en las marcas visuales.

Poses	Ángulo $\beta$	Ángulo $\phi$	Ángulo $\alpha$
$K_0, K_1$	-333.1995	-179.6382	135.6526
$K_1, K_2$	-27.3546	179.1573	137.0036
$K_2, K_3$	314.8888	359.2895	-45.5798
$K_3, K_4$	-153.8592	180.1030	90.0000
$K_4, K_5$	107.8133	180.9404	134.2157
$K_5, K_6$	-109.4857	177.9132	136.6385
$K_6, K_7$	-44.6447	180.0743	90.0000
$K_7, K_8$	-44.2662	180.0539	44.0250
$K_8, K_9$	-72.5303	-2.0207	-44.0250
$K_9, K_{10}$	-244.5054	-178.8387	90.0000
$K_{10}, K_{11}$	-90.9422	-1.0555	-55.5448

Los resultados de este experimento pueden verse en la tabla 2. Estos resultados ofrecen la mayor variación entre las poses  $K_5$  y  $K_6$ , donde el ángulo  $\phi$  difiere del valor real en 2,1 grados. Sin embargo, la media del valor real frente al obtenido en simulación se encuentra por debajo de 1 grado de diferencia, por lo que se puede concluir que se representa con un alto grado de acierto la trayectoria original seguida por el robot.

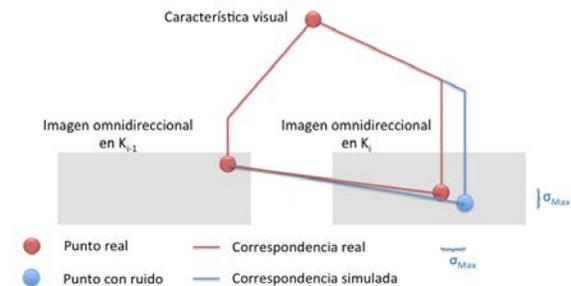


Figura 5: Ruido introducido en la simulación de la imagen en la pose actual ( $K_i$ ) del robot a lo largo de la trayectoria para cada marca o característica visual.

## 5. CONCLUSIONES

En este trabajo, se ha presentado un modelo para resolver la localización de un robot en un entorno

con 4 GDL salvo un factor de escala empleando imágenes omnidireccionales

Se ha reducido el problema a la estimación de la posición y orientación de un conjunto finito de imágenes omnidireccionales. Cada imagen omnidireccional, renombrada como vista, tiene asociado un conjunto de puntos de interés y sus correspondientes descriptores visuales que describen el entorno de una forma compacta. La aportación fundamental se basa en la posibilidad de extraer una transformación entre dos imágenes omnidireccionales en las que existe un conjunto suficiente de correspondencias puntuales, cuando el movimiento ha producido no sólo un cambio en la posición del robot en el plano del suelo, sino que también ha sufrido una variación de altura. La obtención de esta transformación, compuesta por una rotación y una traslación (salvo un factor de escala), permite conocer la pose relativa del robot respecto a una pose anterior, hecho que permite su localización. Se han presentado resultados obtenidos en entornos simulados que validan el trabajo realizado.

Como futuras líneas de investigación se propone resolver el cálculo del factor de escala aplicando únicamente la información obtenida de la imagen omnidireccional. Se plantea así mismo, el estudio de diversos detectores que puedan mejorar los tiempos de computo en la detección de características sin afectar a la robustez de las correspondencias entre imágenes. Y por último, llevar a cabo experimentos con datos reales que confirmen los buenos resultados obtenidos en simulación e integrarlo en un algoritmo de SLAM [3, 7].

### Agradecimientos

Este trabajo ha sido financiado por la Generalitat Valenciana bajo las ayudas del programa AICO de la Consellería de Educación, Investigación, Cultura y Deporte con Ref: AICO/2015/021.

### Referencias

- [1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proceedings of the ninth European Conference on Computer Vision*, May 2006.
- [2] R. Bunschoten and B. Kröse. Visual odometry from an omnidirectional vision system. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 1, pages 577–583 vol.1, Sept 2003.
- [3] A. J. Davison and D. W. Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865–880, Jul 2002.
- [4] Arturo Gil, Oscar Martinez Mozos, Monica Ballesta, and Oscar Reinoso. A comparative evaluation

of interest point detectors and local descriptors for visual slam. *Machine Vision and Applications*, 21(6):905–920, 2010.

- [5] Arturo Gil, Óscar Reinoso, Mónica Ballesta, Miguel Juliá, and Luis Payá. Estimation of visual maps with a robot network equipped with vision sensors. *Sensors*, 10(5):5209, 2010.
- [6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [7] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.
- [8] Jorge J. Moré. *The Levenberg-Marquardt algorithm: Implementation and theory*, pages 105–116. Springer, Berlin, Heidelberg, 1978.
- [9] H. Nagara, Y. Yagi, and M. Yachida. Wide field of view head mounted display for tele-presence with an omnidirectional image sensor. In *Conference on Computer Vision and Pattern Recognition Workshop*, Toronto, Canada, 2003.
- [10] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1988.
- [11] D. Scaramuzza, A. Martinelli, and R. Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, pages 45–45, Jan 2006.
- [12] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor. Omni-directional vision for robot navigation. In *Omnidirectional Vision, 2000. Proceedings. IEEE Workshop on*, pages 21–28, 2000.