

LOCALIZACIÓN DE ROBOTS MÓVILES EN ENTORNOS INDUSTRIALES USANDO UN ANILLO DE CÁMARAS

Sara Roos Hoefgeest Toribio, Álvaro Fernández García, Ignacio Álvarez García, Rafael Corsino González de los Reyes

Área de Ingeniería de Sistemas y Automática. Universidad de Oviedo
Campus Universitario de Gijón, s/n.
{roossara, fernandezgalvaro, ialvarez, rcgonzalez}@uniovi.es

Resumen

Se pretende desarrollar un robot móvil capaz de inspeccionar y reparar chapas de acero en un entorno industrial de grandes dimensiones. Uno de los principales problemas a resolver es la localización del mismo mientras navega sobre la chapa.

El presente documento propone un sistema de localización de un robot móvil basado en la instalación de marcadores ArUco y el uso de un anillo de 8 cámaras calibradas dispuesto sobre el robot que permiten una visión de 360°.

Es preciso conocer la posición de los marcadores respecto a un sistema de coordenadas común. Por ello, se propone una forma de localizar los marcadores y crear un mapa de los mismos con una sola cámara, de tal manera que, posteriormente, pueda ser utilizado para desplazarse sobre la chapa con la mayor precisión posible.

La estrategia escogida se desarrolló en forma de paquetes de ROS capaces de proporcionar el estado del robot a otros algoritmos encargados de tareas como la navegación.

Palabras clave: Robótica, Visión 3D, Localización, Marcadores ArUco, Mapeo, Entorno industrial, Anillo cámaras, ROS.

1 INTRODUCCIÓN

Se desea desarrollar un robot capaz de inspeccionar la totalidad de una chapa de acero en busca de defectos y, posteriormente, ser capaz de subsanarlos. Para ello, el robot deberá ser capaz de navegar con precisión en un entorno industrial de grandes dimensiones.

Uno de los principales problemas es la localización del robot dentro del entorno. A priori, no se puede garantizar que existan elementos externos que puedan ser utilizados como balizas. Para resolver este

problema, se propone un sistema de localización basado en la instalación de marcadores ArUco y el uso de un anillo de 8 cámaras dispuesto sobre el robot móvil.

No se puede garantizar que la posición exacta de los marcadores con respecto a un sistema de coordenadas común sea conocida. En este artículo se propone una manera de localizar los marcadores y elaborar un mapa de los mismos utilizando una sola cámara, que permita, posteriormente, ser utilizado para que el robot se desplace sobre la chapa con la mayor precisión posible.

El sistema de mapeo utiliza una cámara calibrada para estimar la pose de los marcadores respecto a un sistema de referencia común, elegido en la posición de uno de los marcadores en el entorno. Una vez realizada una primera estimación de sus posiciones se aplica un ajuste *bundle* para minimización del error y optimizar sus localizaciones.

En cuanto a la localización, cada cámara que forma el arco sigue, individualmente, un esquema monocular, estimando la posición del robot resolviendo el problema de *Perspective-n-Point* (PnP) a partir de las poses 3D de marcadores y sus proyecciones en sus imágenes. Posteriormente, se escoge la mejor estimación entre las 8 cámaras y se someten los datos a un filtro de Kalman para eliminar malas estimaciones y conseguir una localización más fiable.

2 ESTADO DEL ARTE

Cada vez resulta más habitual encontrar robots móviles en las industrias. La localización en cada instante de tiempo es vital para estos robots. Este problema se ha tratado desde múltiples perspectivas, siendo común el uso de diferentes sensores que proporcionan información directa sobre la localización del robot en cada instante, o bien, sobre los cambios que se han producido en su entorno.

Se puede realizar una primera clasificación de estas técnicas en dos grupos según se utilicen medidas relativas, llamado odometría, o medidas absolutas, conocido simplemente como localización global.

Si la posición inicial del robot es conocida, los métodos odométricos estiman la posición y orientación del vehículo mediante información obtenida de diferentes sensores incluidos en el robot, como encoders, giroscopios o acelerómetros, que permiten la estimación incremental de la pose actual a partir de la anterior. Al tratarse de un método incremental se van acumulando errores tras cada iteración, por tanto, es necesario aplicar continuos ajustes durante la navegación.

Los métodos globales obtienen la pose absoluta a partir de balizas y marcadores o señales satélite, como los sistemas GPS. Sin embargo, los sistemas basados en señales satélite no resultan adecuados en espacios confinados ya que, los GPS no reciben señal en recintos cerrados.

Una estrategia para conocer la pose de un vehículo utilizando medidas absolutas es mediante la colocación de balizas en posiciones conocidas del entorno con el fin de facilitar la localización del robot. Esta técnica es una de las más precisas en este ámbito. Si bien, su precisión está estrechamente relacionada con el tipo de señal utilizada, por ejemplo, radio [20], láser [10], Wi-Fi [3], infrarrojos [12] o ultrasonidos [10], con las características del sensor y con el número de balizas empleadas. También existen balizas visuales, como balizas luminosas de diferentes geometrías [9] y colores o marcadores fiduciales [2].

Existen otras alternativas como el *Simultaneous Localization and Mapping* o SLAM, que permite, junto a la localización, la construcción simultánea de un mapa del entorno, [5], [18], [26]. También es frecuente en robótica la fusión de sensores, generalmente mediante filtros de Kalman [16] o filtros de partículas [4], que permite combinar las estimaciones procedentes de distintas fuentes para lograr una pose más robusta. Por ejemplo, en [1], se fusiona la información procedente de sensores inerciales con un método de odometría visual.

En este documento se expone un método de localización mediante marcadores artificiales. En este sentido, se pueden encontrar diferentes sistemas de marcadores fiduciales, compuestos por un conjunto de marcadores definidos y un algoritmo que permite su detección e identificación. Los sistemas basados en marcadores cuadrados son los más populares, debido a que la estimación de la posición de una cámara calibrada es posible a partir de sus cuatro esquinas. Estos marcadores presentan un identificador único basado en un código binario, generalmente

codificados dentro de un diccionario que almacena un conjunto de marcadores previamente desarrollados.

Los marcadores binarios han sido utilizados en múltiples aplicaciones de localización de vehículos. Por ejemplo, investigadores del grupo AVA, de la Universidad de Córdoba, han estado trabajando en este ámbito desarrollando diferentes aplicaciones para localizar robots móviles utilizando marcadores artificiales tipo ArUco desplegados por el entorno con una sola cámara[24].

Otros sistemas de marcadores cuadrados destacados son ARTag [6], ARToolKit Plus [21] o BinARyID [7]. Entre los marcadores no cuadrados, los más empleados son los circulares Intersense [13] y ReacTIVision [27], con forma de ameba.

3 MÉTODOS

En este artículo se propone un enfoque de localización global que estima la pose del robot a partir de una serie de marcadores dispuestos en el entorno.

Previamente a la localización del robot, es preciso realizar un mapeo de los marcadores para obtener sus poses respecto a un sistema de referencia global. En ese sentido, se desarrolló un sistema de mapeado de marcadores a partir de una serie de imágenes de los mismos, tomadas con una sola cámara calibrada.

El sistema de referencia global, al que se llamará sistema del mundo, se fijará en el lugar del primer marcador detectado en las imágenes que cumpla unas determinadas características. Una vez localizado el primer marcador y fijado el sistema del mundo, se irán estimando las poses del resto de los marcadores cuando se detecten en las imágenes junto a marcadores previamente almacenados en el mapa.

Para conseguir realizar el mapeo satisfactoriamente, es necesario que, cada vez que un nuevo marcador sea detectado, esté presente en la misma imagen otro cuya pose ya haya sido estimada previamente.

Una vez conocida la localización de los marcadores, se estima la posición y orientación del vehículo a partir de los marcadores encontrados mediante un arco de 8 cámaras calibradas dispuestas en el robot.

La estrategia de localización implementada comienza con una etapa de detección de los marcadores en las imágenes de cada una de las cámaras que conforman el arco. Después, cada cámara sigue un enfoque monocular calculando la pose absoluta del robot a partir de las correspondencias de las proyecciones 2D de las esquinas de los marcadores y sus

correspondientes puntos 3D almacenados en el mapa de marcadores. Finalmente, se escoge la mejor estimación entre las 8 cámaras y se aplica un Filtro de Kalman para descartar malas estimaciones y proporcionar una localización más fiable.

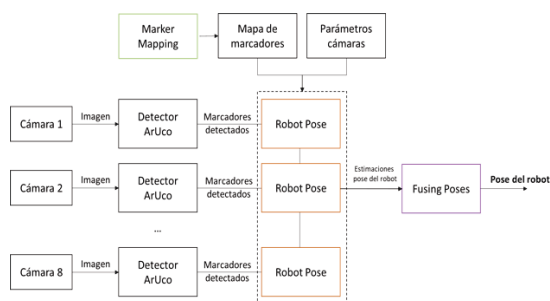


Figura 1. Esquema del método propuesto. *Robot Pose* hace referencia al paquete de localización para cada una de las cámaras. *Fusing Poses* junta la información de las 8 cámaras para proporcionar la pose final del robot. *Marker Mapping* es el paquete de mapeo de marcadores.

3.1 MAPEADO DE LOS MARCADORES

Para llevar a cabo la localización del robot utilizando marcadores dispuestos en el entorno es preciso conocer sus poses respecto a un sistema de referencia común. Para ello, se desarrolló un sistema de mapeo de marcadores a partir de una serie de imágenes de los mismos proporcionadas por una sola cámara calibrada.

Para que la cámara pueda capturar todos los marcadores el robot deberá moverse. En ese sentido, el robot realizará un movimiento controlado que asegure la visión de todos los marcadores. El movimiento del robot podrá alterarse en función del entorno para permitir la visión de todos los marcadores.

En el mapa se almacenan las poses de los marcadores referidas a un sistema de referencia común, al que se llamará sistema del mundo, que se fijará en uno de los marcadores ArUco.

El algoritmo de mapeo desarrollado se puede dividir en una serie de etapas. En primer lugar, se detectarán los marcadores en las imágenes y se escogerá uno de ellos para fijar el sistema de coordenadas del mundo. Posteriormente, se estimarán las poses del resto de marcadores respecto del sistema de referencia. Finalmente, una vez tenida una estimación de las poses, se aplicará un ajuste *bundle* para minimizar el error de la estimación y optimizar las poses de los marcadores.

El marcador que definirá la posición y orientación del sistema de coordenadas global será aquel que esté más cercano a la cámara, tenga una orientación más favorable y además tenga otros marcadores más cerca.

El sistema de coordenadas asociado a cada marcador se puede ver en la Figura 2 (b), en el que el eje Z será perpendicular al plano que contiene al marcador y el origen situado en el centro del mismo.

Una vez localizado el primer marcador y fijado el sistema del mundo, se irán estimando las poses del resto de los marcadores cuando se detecten en las imágenes junto a marcadores previamente almacenados en el mapa. Para ello, se recorren las imágenes que presenten más de un ArUco estimando la pose relativa entre ellos y con la cámara.

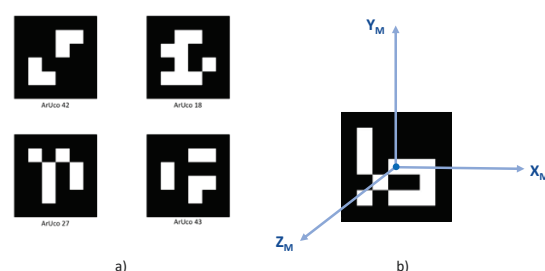


Figura 2. a) Marcadores ArUco b) Sistema de coordenadas de un marcador.

3.1.1 Localización de los marcadores ArUco

El sistema de marcadores ArUco [22] está formado por un conjunto de marcadores y un algoritmo que permite su detección e identificación.

Un marcador ArUco, ver Figura 2, es un marcador sintético cuadrangular compuesto por un borde negro y una matriz binaria interna que determina un identificador. El tamaño del marcador determina el tamaño de la matriz interna. Existen diferentes diccionarios ArUco que agrupan marcadores con distintos identificadores.

A priori, los únicos datos de los que se dispone son: el número de marcadores dispuestos en el entorno, sus identificadores y el tamaño de los mismos. A partir de estos datos y las imágenes tomadas por la cámara se detectan los marcadores en las imágenes utilizando un paquete de ROS ya desarrollado, *aruco_detect* [23]. Este paquete utiliza las librerías que proveen los autores del sistema de marcadores, explicado en el artículo original [8]. El algoritmo proporciona las 4 esquinas y el identificador de cada marcador presente en la imagen.

Para calcular la pose de los marcadores respecto del mundo, es preciso obtener las transformaciones entre

los sistemas de la cámara para cada imagen tomada y los marcadores presentes en las mismas.

Para modelar la cámara se supone un modelo *pin-hole* [17]. Resolviendo el problema de *Perspective-n-Point* (PnP) [11] es posible estimar la pose de una cámara calibrada dados una serie de puntos 3D en el mundo y sus correspondientes proyecciones 2D en la imagen. Un solo marcador arroja las cuatro correspondencias necesarias, a partir de sus esquinas, para resolver el problema. La pose de la cámara consiste en 6 grados de libertad, orientación (*roll*, *pitch*, *yaw*) y traslación respecto del mundo (X, Y, Z).

Resolviendo el problema PnP dadas las posiciones 3D de las esquinas en el sistema del propio marcador y sus correspondientes proyecciones 2D en la imagen, se obtiene la pose de la cámara calibrada respecto del sistema del propio marcador.

La figura 3 muestra este proceso. I1 e I2 representan las imágenes tomadas por la cámara que hace el mapeo para las posiciones 1 y 2 del robot. Supongamos que en la primera posición seleccionada se ven los marcadores 0 y 1, mientras que en la segunda se ven los marcadores 1 y 2. Usando los algoritmos descritos en los párrafos anteriores, obtendremos las transformaciones entre los sistemas de coordenadas de cada marcador y la posición de la cámara en cada imagen. En particular, para la imagen 2 obtendremos las transformaciones ${}^{C2}T_{M1}$ y ${}^{C2}T_{M2}$. Al ser común la posición de la cámara, se puede calcular la transformación entre ambos marcadores. En particular, la transformación que refiere la posición del marcador 2 respecto al marcador 1 será:

$${}^{M1}T_{M2} = {}^{M1}T_{C2} \cdot {}^{C2}T_{M2} \tag{1}$$

Así, si uno de los dos marcadores ya ha sido almacenado en el mapa, la transformación del marcador desconocido respecto del mundo se obtiene de una manera directa. En este ejemplo, se supone que el sistema asociado al marcador 0 se toma como sistema de coordenadas del mundo. Usando la ecuación (1), podemos calcular la transformación existente entre el marcador 1 y el sistema de coordenadas global, puesto que los marcadores 0 y 1 son visibles en la misma imagen. Así, conoceremos ${}^WT_{M1} = {}^{M0}T_{M1}$. Para calcular la pose del sistema de coordenadas asociado al marcador 2 respecto al sistema de mundo bastará con aplicar la ecuación (2) y así podremos añadirlo al mapa

$${}^WT_{M2} = {}^WT_{M1} \cdot {}^{M1}T_{M2} \tag{2}$$

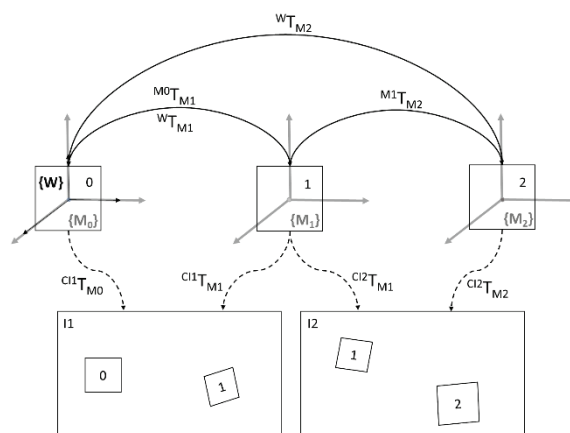


Figura 3. Esquema transformaciones del sistema de mapeo. Siendo $\{M_0\}$, $\{M_1\}$, $\{M_2\}$ los sistemas de coordenadas de los marcadores en el espacio e I1, I2 dos imágenes tomadas por la cámara en instantes diferentes.

En el caso de haber más de un marcador almacenado en el mapa, se calcula la pose del marcador desconocido respecto al marcador conocido más cercano a la cámara, previsiblemente mejor detectado.

Una vez obtenida una primera estimación de la localización de todos los marcadores presentes en el entorno, se aplica un *ajuste bundle* [19] para optimizar las poses.

3.1.2 Ajuste bundle

La solución propuesta consiste en buscar una optimización global de la posición 3D de los marcadores mediante la minimización del error de reproyección de las esquinas de todos los marcadores presentes en las imágenes mediante un *ajuste bundle*.

Este método consiste en refinar conjuntamente un conjunto de estimaciones iniciales de los parámetros de la cámara para encontrar el conjunto de parámetros que predice con mayor precisión las ubicaciones de los puntos 3D observados en las imágenes disponibles buscando minimizar el error de reproyección de los mismos.

Dada una estimación inicial de las matrices de proyección de las diferentes vistas P_i , y un conjunto de puntos 3D reconstruidos con su proyección 2D en la imagen $\{X_j, x_j^i\}$ se busca minimizar el error de reproyección con respecto a todos los puntos 3D y los parámetros de la cámara:

$$\sum_j \sum_{i \in V(j)} |P_i(X_j) - x_j^i|^2 \tag{3}$$

Donde $V(j)$ es la lista de índices de las vistas de la cámara en los que el punto X_j es visible. $P_i(X_j)$ representa la reproyección 2D del punto X_j en la cámara i utilizando la matriz de proyección P_i .

Un método eficiente para resolver este problema es la minimización de Levenberg-Marquart [15]. Se trata de un método iterativo utilizado para resolver problemas de ajuste de métodos por mínimos cuadrados no lineales.

3.2 LOCALIZACIÓN DEL ROBOT

En primer lugar, se detectan los marcadores presentes en las imágenes capturadas por las cámaras en cada instante. Como ya se ha explicado previamente, la detección de los marcadores en las imágenes se lleva a cabo utilizando el paquete *aruco_detect* [23]. Este paquete proporciona las 4 esquinas y el identificador de cada marcador presente en la imagen.

Una vez detectados los marcadores en las imágenes, se realiza una estimación de la pose del robot a partir los presentes en cada una de las cámaras.

Para modelar las cámaras se supone un modelo *pin-hole* [17]. Resolviendo el problema de *Perspective-n-Point* (PnP) [11] es posible estimar la pose de una cámara calibrada dados una serie de puntos 3D en el mundo y sus correspondientes proyecciones 2D en la imagen. Un solo marcador arroja las cuatro correspondencias necesarias, a partir de sus esquinas, para resolver el problema. La pose de la cámara consiste en 6 grados de libertad, orientación (*roll, pitch, yaw*) y traslación respecto del mundo (X, Y, Z).

Se resuelve el problema PnP con las proyecciones de las cuatro esquinas de los marcadores, proporcionadas por el paquete de detección de ArUcos, y las posiciones 3D de los marcadores correspondientes respecto del sistema del mundo, obtenidas del mapa de marcadores calculado previamente.

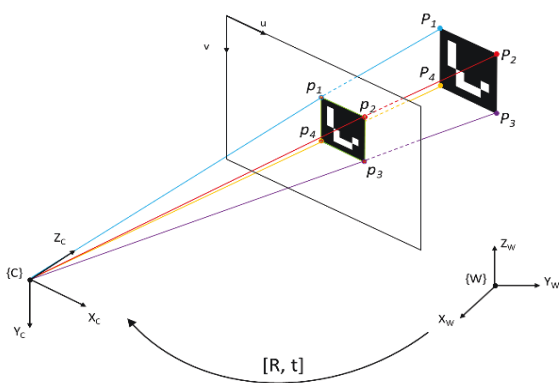


Figura 4. Estimación de la pose de la cámara a partir de un marcador. ${}^C T_W = [{}^C R_W \mid {}^C t_W]$

Tras resolver el problema, se obtiene la transformación entre el sistema de coordenadas de la cámara y el sistema del mundo, ${}^C T_W$, ver Figura 4. De esta manera, obtener la matriz de transformación del robot y el sistema del mundo ${}^W T_R$, se puede calcular mediante de manera directa como:

$${}^W T_R = {}^W T_C \cdot {}^C T_R \tag{4}$$

Donde ${}^W T_C$ es la transformación inversa de ${}^C T_W$, y ${}^C T_R$, la transformación que relaciona los sistemas de coordenadas de la cámara y el robot, definido en la fase de montaje del arco de cámaras sobre el mismo.

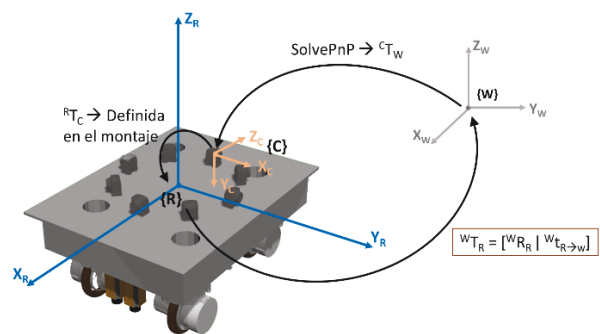


Figura 5. Transformaciones entre sistemas de referencia.

Para cada una de las cámaras se obtiene una estimación de la pose del robot en función de los marcadores presentes en cada una. Con el fin de escoger la mejor localización, se realiza una estimación de la precisión de la medida, basada en el error de reproyección y la distancia a la que el marcador se encuentra de la cámara.

Así, las estimaciones asociadas al menor error entre las ocho cámaras en cada instante de tiempo son sometidas a un filtrado de eliminación de *outliers*, descartando la medida si difiere notablemente de las anteriores o tiene un valor no coherente con el comando de movimiento ejecutado por el robot.

Tras pasar esa primera medida de eliminación de *outliers*, se incorpora la estimación a un Filtro de Kalman Lineal con el fin de hacer la localización final más robusta.

3.2.1 Filtro de Kalman Lineal

El filtro de Kalman [14] es un filtro recursivo de predicción que se basa en el uso de técnicas de espacio de estados para estimar el estado de un sistema

dinámico. En este proyecto, se ha implementado un filtro de Kalman lineal para eliminar malas estimaciones y conseguir una localización más fiable.

El filtro de Kalman se divide en dos etapas principales, predicción y corrección. En el primer paso, el modelo dinámico predice el estado del sistema. En el segundo, se corrige dicha predicción con el modelo de observación. Este procedimiento se repite para cada intervalo de tiempo.

Los estados sucesivos $s_t \in R^n$ de un proceso controlado se relacionan con un modelo dinámico que describe la transformación del vector de estado en el tiempo, descrito en la ecuación (5).

$$s_t = As_{t-1} + w_t \tag{5}$$

Siendo A la matriz de estados y w_t el ruido del proceso. El vector de estado que compone la matriz se definirá con un total de 18 estados (ecuación (6)): La posición (x,y,z), con su primer y segunda derivadas, es decir, velocidad y aceleración. La información de rotación en forma de ángulos de Euler (roll (ψ), pitch (θ), yaw (ϕ)), con su primer y segunda derivada, correspondientes a velocidad y aceleración angular.

$$X = (x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}, \psi, \theta, \phi, \dot{\psi}, \dot{\theta}, \dot{\phi}, \ddot{\psi}, \ddot{\theta}, \ddot{\phi}) \tag{6}$$

La medida de la posición del robot en cada instante t, se relaciona con el estado s_t mediante el modelo de observación lineal de la ecuación (7), donde v_t representa el ruido de la medida y C la matriz de observación.

$$z_t = Cs_t + v_t \tag{7}$$

El vector de medidas se compone de 6 medidas, correspondientes a la posición (x,y,z) y orientación, en forma de ángulos de Euler (ψ, θ, ϕ), del robot.

Como se mencionó previamente, la primera etapa del filtro de Kalman es la predicción. En cada instante de tiempo se realiza una primera estimación del estado actual, denominado estado a priori s_t^- , calculado al despreciar el ruido dinámico y resolviendo las ecuaciones que describen el modelo dinámico, ecuación (8). Su matriz de covarianzas S_t^- es calculada durante esta etapa según la ecuación (9), donde S_{t-1} es la covarianza del error de la estimación a posteriori en el instante anterior y Λ_w es la covarianza del ruido del proceso que mide la calidad del modelo de movimiento respecto a la realidad

$$s_t^- = As_{t-1} \tag{8}$$

$$S_t^- = AS_{t-1}A^T + \Lambda_w \tag{9}$$

Posteriormente, sucede la etapa de corrección. En ella se mejora la predicción s_t^- con las observaciones realizadas en el instante t , (z_t), obteniéndose la estimación a posteriori s_t según la ecuación (10). Donde la diferencia ($z_t - z_t^-$) es la medida residual que refleja la diferencia entre la medida prevista $z_t^- = Cs_t^-$ y la media real z_t .

En esta ecuación, el estado estimado y las mediciones son ponderados para calcular el estado corregido según G_t , matriz de ganancia del filtro. G_t es calculada según la ecuación (12), siendo Λ_v la matriz de covarianza de las medidas.

Su matriz de covarianza S_t es calculada según la ley de propagación del error, ecuación (11).

$$s_t = s_t^- + G_t(z_t - z_t^-) \tag{10}$$

$$S_t = S_t^- - G_tCS_t^- \tag{11}$$

$$G_t = S_t^-C^T(CS_t^-C^T + \Lambda_v)^{-1} \tag{12}$$

4 EXPERIMENTOS Y RESULTADOS

La estrategia escogida se desarrolló en forma de paquetes de ROS capaces de proporcionar el estado del robot a otros algoritmos encargados de tareas como la navegación.

La detección de marcadores se ha llevado a cabo utilizando un paquete ya desarrollado, *aruco_detect* [23], que toma como base las librerías originales del sistema ArUco [8].

Por otro lado, se han desarrollado paquetes tanto del sistema de localización como del de mapeo de marcadores según los algoritmos expuestos anteriormente.

Se llevaron a cabo diferentes experimentos con el fin de evaluar los sistemas desarrollados. Las pruebas se desarrollaron sobre el sistema operativo Ubuntu 16.04 LTS y ROS (*Robot Operating System*) [28], que proporciona las librerías y herramientas necesarias para implementar aplicaciones de robótica. En concreto, se utilizó la versión ROS Kinetic.

Además, los experimentos se llevaron a cabo en distintas simulaciones realizadas en el programa Gazebo [25], recreando un entorno industrial que pretende emular las instalaciones por las que se moverá el robot, se puede ver en la Figura 7 (a).

4.1 MAPEO DE ARUCOS

Se realizaron pruebas en diferentes escenas para evaluar el sistema de mapeo de marcadores. En primer lugar, se evaluó en una escena con 8 marcadores, de 17 cm de lado, cercanos entre sí formando un círculo de diámetro 4 metros alrededor del robot, ver Figura 6. En este caso, el robot realizó un movimiento circular tomando un total de 9 imágenes.

Posteriormente, se evaluó el sistema en una escena industrial con una chapa en el centro para inspeccionar. En este caso, los marcadores se situaron alrededor de una chapa, de dimensiones 14x5m, con el objetivo de permitir al robot localizarse al inspeccionar y reparar la misma, ver Figura 7. Se tomaron un total de 33 imágenes.

A raíz de las figuras anteriores, se puede apreciar que los resultados obtenidos se ajustan a la disposición real. En el primer experimento los errores, recogidos en la Tabla 1, son menores, esto es debido a que los marcadores se encuentran más cerca de la cámara y

están más cerca unos de otros, lo que facilita la detección y el cálculo de las transformaciones. Sin embargo, en el experimento de la figura 7, los errores aumentan, ver Tabla 2, debido a una mayor lejanía de los marcadores con la cámara y entre sí, lo que obliga a que los marcadores se encuentren lo suficientemente lejos como para poder capturar un mínimo de dos en la misma imagen.

En estos experimentos, la optimización mediante el ajuste *bundle* no ha supuesto mejoría, siendo los errores prácticamente iguales que en la estimación previa al ajuste. Esto es debido a que estamos trabajando mediante simulación con cámaras calibradas en un caso ideal. Por ello, los errores de reproyección ya son pequeños en un inicio y es difícil minimizarlos aún más. En un futuro, cuando se trabaje con cámaras reales, se espera que el ajuste *bundle* sí optimice la posición de los marcadores. Los errores obtenidos en estos experimentos se pueden asociar a errores en la detección de los marcadores en las imágenes.

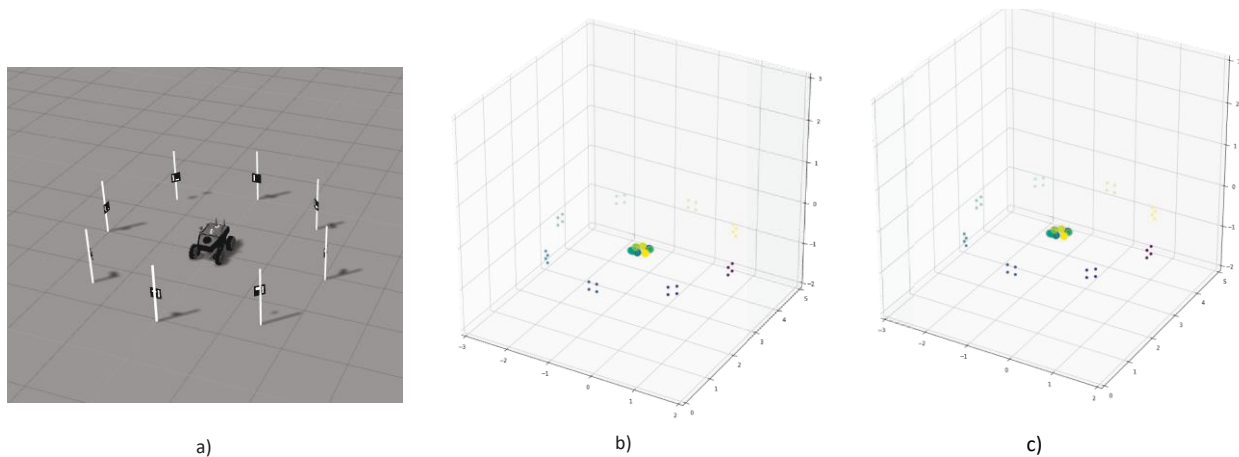


Figura 6. Resultados mapeo marcadores. a) Escena simulada en Gazebo. b) Primera estimación de las posiciones de los marcadores. c) Resultado tras ajuste *bundle*. En b) y c) los puntos representan las esquinas de los marcadores, y los círculos las posiciones de las cámaras, codificados por color.

Tabla 1. Errores obtenidos en el experimento de la Figura 6. X,Y,Z representan las estimaciones previas al ajuste *bundle* e X_{ba}, Y_{ba}, Z_{ba} , las estimaciones tras el ajuste.

	X	Y	Z	X_{ba}	Y_{ba}	Z_{ba}
Error medio (m)	0.043	0.038	0.063	0.042	0.038	0.064
Error máximo (m)	0.158	0.104	0.212	0.158	0.106	0.210
Error mínimo (m)	0.001	0.001	0.004	0.001	0.004	0.002

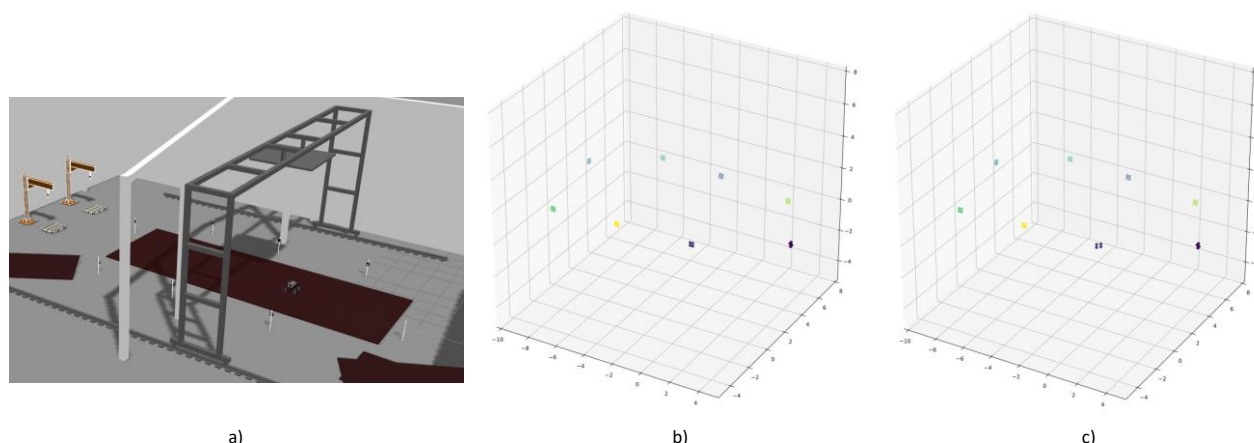


Figura 7. Resultados mapeo marcadores. a) Escena simulada en Gazebo. b) Primera estimación de las posiciones de los marcadores. c) Resultado tras ajuste *bundle*. En b) y c) los puntos representan las esquinas de los marcadores, codificados por color.

Tabla 2. Errores obtenidos en el experimento de la Figura 7. X,Y,Z representan las estimaciones previas al ajuste *bundle* e X_{ba}, Y_{ba}, Z_{ba} , las estimaciones tras el ajuste.

	X	Y	Z	X_{ba}	Y_{ba}	Z_{ba}
Error medio (m)	0.110	0.245	0.180	0.111	0.247	0.192
Error máximo (m)	0.323	0.968	0.444	0.324	0.967	0.444
Error mínimo (m)	0.003	0.005	0.012	0.001	0.001	0.009

4.2 LOCALIZACIÓN

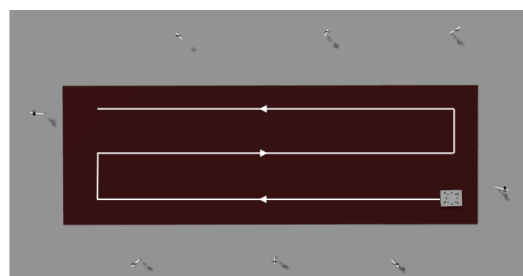
Con el fin de evaluar el método de localización propuesto, se desarrolló una simulación en la que el robot realiza una trayectoria sobre la chapa, de dimensiones 14x5m, simulando inspeccionarla. Se distribuyeron 8 ArUcos de 17 cm de lado alrededor de la misma.

En la Figura 8 y Figura 9 se muestran los resultados de la localización. En particular, las componentes de la pose más importantes son, las posiciones en los ejes X e Y y el ángulo *yaw*, giro respecto del eje Z, ya que el robot siempre se va a mover sobre el plano X-Y.

A partir de la Figura 8 y los datos expuestos en la Tabla 3, se puede apreciar que la trayectoria estimada se ajusta bastante a la real. El máximo error alcanzado ha sido de 47 centímetros, sin embargo, los errores habituales son bastante más bajos, teniendo un error

Tabla 3. Resultados localización

	X	Y
Error medio (m)	0.146	0.039
Error máximo (m)	0.479	0.270
Error mínimo (m)	0.0001	0.0001



medio en la posición en X e Y de 14cm y 3cm respectivamente. Se puede observar que dónde mayor error hay en la trayectoria es en los extremos, precisamente donde hay un menor número de marcadores cercanos.

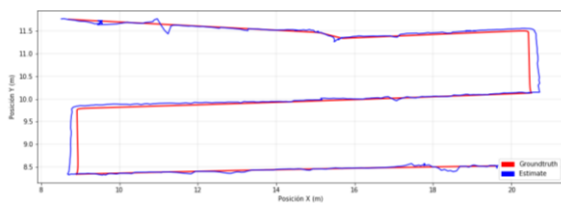


Figura 8. Arriba: Trayectoria seguida en la simulación emulando un proceso de inspección de chapa. Abajo: Resultados. (En rojo el *groundtruth* y en azul la trayectoria estimada)

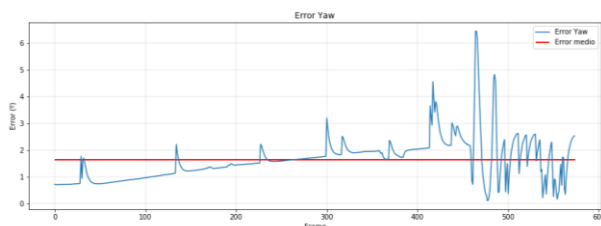


Figura 9. Error *yaw* en la trayectoria de la Figura 8.

La orientación del robot también ha sido estimada de una manera bastante fiable, ver figura anterior. El error medio está en torno a 1.8° , siendo el máximo error igual a 6.2° .

5 DISCUSIÓN Y CONCLUSIONES

En este artículo se ha propuesto un método de localización de un robot móvil en un entorno industrial utilizando un arco de 8 cámaras, que permiten una visión de 360° , y marcadores ArUco. Además, se ha propuesto un algoritmo de mapeado capaz de localizar los marcadores respecto a un sistema de referencia común.

Tras evaluar los diferentes algoritmos propuestos se puede afirmar que los resultados han sido satisfactorios.

Sin embargo, esta estrategia presenta ciertas limitaciones. Si los marcadores son correctamente detectados el sistema proporciona resultados satisfactorios. Sin embargo, si los marcadores se encuentran lejos de la cámara, la detección arrojará peores resultados y, por lo tanto, el mapeo de los marcadores y la localización del robot será peor.

Se abren diferentes vías de trabajo futuro. Actualmente, el sistema de mapeo trabaja con una única cámara. En futuras actualizaciones se extenderá el método para trabajar con todas las cámaras del arco, permitiendo añadir restricciones en cuanto a las posiciones de las cámaras en las ecuaciones del ajuste *bundle*. Además, también se podrían añadir restricciones para optimizar las cuatro esquinas de cada marcador conjuntamente.

En un futuro, se estudiará si la localización mediante balizas se podría fusionar con información procedente de otras fuentes, como una unidad de medición inercial (IMU) o las órdenes de movimiento procedentes del sistema de navegación. Por ejemplo, mediante un filtro de Kalman extendido, que permite la fusión de información de diferentes fuentes con el fin de estimar la pose del robot con un menor error.

Los experimentos han sido realizados mediante simulación debido a que aún no se disponía de un modelo físico. En un futuro, se pretende evaluar, empleando un robot real, el funcionamiento de los algoritmos desarrollados.

English summary

MOBILE ROBOT LOCALIZATION IN INDUSTRIAL ENVIRONMENTS USING A CAMERA RING

Abstract

The goal of this project is to develop a mobile robot capable of inspecting and repairing steel sheets in an industrial environment.

This document proposes a localization system of a mobile robot base on ArUco markers and a ring formed by 8 cameras. The ring is mounted on top of the robot so that provides a 360° vision.

Previously to the localization process, it is necessary to map the pose of ArUco markers. In this way, a mapping system has been developed that estimates the poses of the markers with respect to a common reference system, attached to the first marker found.

We have developed ROS Kinetic packages, to solve the marker mapping problem and are able to provide the localization of the robot to other modules, such as navigation.

Keywords: Robotics, 3D Vision, Robot localization, ArUco markers, Mapping, Industrial Environment, Multi-camera ring, ROS.

Referencias

1. Alatise, M., Hancke, G.: Pose Estimation of a Mobile Robot Based on Fusion of IMU Data and Vision Data Using an Extended Kalman Filter. *Sensors*. 17, 10, 2164 (2017). <https://doi.org/10.3390/s17102164>.
2. Babinec, A. et al.: Visual Localization of Mobile Robot Using Artificial Markers. *Procedia*

- Engineering. 96, 1–9 (2014).
<https://doi.org/10.1016/j.proeng.2014.12.091>.
3. Biswas, J., Veloso, M.: WiFi localization and navigation for autonomous indoor mobile robots. In: 2010 IEEE International Conference on Robotics and Automation. pp. 4379–4384 IEEE, Anchorage, AK (2010).
<https://doi.org/10.1109/ROBOT.2010.5509842>.
 4. Caron, F. et al.: Particle Filtering for Multisensor Data Fusion With Switching Observation Models: Application to Land Vehicle Positioning. IEEE Transactions on Signal Processing. 55, 6, 2703–2719 (2007). <https://doi.org/10.1109/TSP.2007.893914>.
 5. Davison, A.J.: SLAM with a Single Camera, https://www.doc.ic.ac.uk/~ajd/Publications/davison_cm2002.pdf.
 6. Fiala, M.: ARTag, a Fiducial Marker System Using Digital Techniques. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). pp. 590–596 IEEE, San Diego, CA, USA (2005).
<https://doi.org/10.1109/CVPR.2005.74>.
 7. Flohr, D., Fischer, J.: A Lightweight ID-Based Extension for Marker Tracking Systems. 7.
 8. Garrido-Jurado, S. et al.: Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition. 47, 6, 2280–2292 (2014). <https://doi.org/10.1016/j.patcog.2014.01.005>.
 9. John J. Leonard, Hugh F. Durrant-Whyte: Mobile Robot Localization by Tracking Geometric Beacons. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION. 7, NO 3, (1991).
 10. Ko, N., Kuc, T.-Y.: Fusing Range Measurements from Ultrasonic Beacons and a Laser Range Finder for Localization of a Mobile Robot. Sensors. 15, 5, 11050–11075 (2015).
<https://doi.org/10.3390/s150511050>.
 11. Lu, X.X.: A Review of Solutions for Perspective-n-Point Problem in Camera Pose Estimation. J. Phys.: Conf. Ser. 1087, 052009 (2018).
<https://doi.org/10.1088/1742-6596/1087/5/052009>.
 12. McGillem, C.D., Rappaport, T.S.: Infra-red location system for navigation of autonomous vehicles. In: Proceedings. 1988 IEEE International Conference on Robotics and Automation. pp. 1236–1238 IEEE Comput. Soc. Press, Philadelphia, PA, USA (1988).
<https://doi.org/10.1109/ROBOT.1988.12230>.
 13. Naimark, L., Foxlin, E.: Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In: Proceedings. International Symposium on Mixed and Augmented Reality. pp. 27–36 IEEE Comput. Soc, Darmstadt, Germany (2002).
<https://doi.org/10.1109/ISMAR.2002.1115065>.
 14. Pi, E.R.: Implementation of a 3D pose estimation algorithm. 67.
 15. Ranganathan, A.: The Levenberg-Marquardt Algorithm. 5.
 16. Söderström, T.: Using an Extended Kalman Filter for Rigid Body Pose Estimation. Journal of Biomechanical Engineering. 127, 3, 475 (2004).
<https://doi.org/10.1115/1.1894371>.
 17. Sonka, M. et al.: Image Processing, Analysis, and Machine Vision.
 18. Tateno, K. et al.: When 2.5D is not enough: Simultaneous reconstruction, segmentation and recognition on dense SLAM. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). pp. 2295–2302 (2016).
<https://doi.org/10.1109/ICRA.2016.7487378>.
 19. Triggs, B. et al.: Bundle Adjustment — A Modern Synthesis. In: Triggs, B. et al. (eds.) Vision Algorithms: Theory and Practice. pp. 298–372 Springer Berlin Heidelberg, Berlin, Heidelberg (2000). https://doi.org/10.1007/3-540-44480-7_21.
 20. Vascak, J., Savko, I.: Radio Beacons in Indoor Navigation. In: 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA). pp. 283–288 IEEE, Kosice (2018).
<https://doi.org/10.1109/DISA.2018.8490529>.
 21. Wagner, D., Schmalstieg, D.: ARToolKitPlus for Pose Tracking on Mobile Devices. 9.
 22. ArUco: a minimal library for Augmented Reality applications based on OpenCV | Aplicaciones de la Visión Artificial,
<https://www.uco.es/investiga/grupos/ava/node/26>.
 23. aruco_detect - ROS Wiki,
http://wiki.ros.org/aruco_detect.
 24. Camera Pose Estimation | Aplicaciones de la Visión Artificial,
<http://www.uco.es/investiga/grupos/ava/node/61>.
 25. Gazebo, <http://gazebo.org/>.
 26. New package: (SVO) Semi-direct Monocular Visual Odometry - ROS robotics news,
<http://www.ros.org/news/2014/06/new-package-svo-semi-direct-monocular-visual-odometry.html>.
 27. reacTIVision,
<http://reactivision.sourceforge.net/#usage>.
 28. ROS.org | Powering the world's robots,
<http://www.ros.org/>.



© 2019 by the authors. Submitted for possible open access publication under the terms and conditions of the

Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).