

Proyecto de Investigación de Excelencia de la  
Junta de Andalucía P10-TEP-6101-R

“Navegación autónoma de un robot móvil 4x4 en  
entornos naturales mediante GPS diferencial y  
telémetro láser tridimensional”

## **Procesado de una nube de puntos para la creación de un mapa de elevación borroso**

Manuel Zafra Granados  
Jorge L. Martínez Rodríguez  
Antonio J. Reina Terol

Departamento de Ingeniería de Sistemas y Automática  
Universidad de Málaga

Diciembre de 2015

# Índice

<b>1. Objetivos</b>	<b>2</b>
<b>2. Punto de partida</b>	<b>2</b>
<b>3. Implementación en ROS</b>	<b>3</b>
3.1. Filtrado . . . . .	3
3.2. Creación de mapas . . . . .	6
<b>4. Resultados experimentales</b>	<b>10</b>
<b>5. Referencias</b>	<b>12</b>

## 1. Objetivos

Una nube de puntos 3D en bruto ofrece poca información útil para un robot móvil y un alto coste en memoria. Por ello, resulta necesario procesar la nube de puntos y extraer información valiosa, a la vez que se reduce su volumen en memoria.

El objetivo principal de este informe es describir la creación de un mapa de elevación que, posteriormente, permita la planificación de un camino para la navegación local de Andábata. Se ha decidido usar un mapa de elevación difuso donde, previamente, se han eliminado los voladizos y puntos muy alejados mediante un filtro. También se debe obtener una máscara de confianza sobre el mapa difuso.

## 2. Punto de partida

Andábata es capaz de obtener periódicamente barridos 3D con su telémetro láser. Estos barridos 3D son publicados con cierta frecuencia y contienen todos los puntos capturados por un telémetro láser 2D mientras da una vuelta completa [1]. El punto de inicio de este documento es dicha nube de puntos, que esta nivelada y referenciada al origen de coordenadas del robot al iniciar el barrido.

### 3. Implementación en ROS

Para alcanzar los objetivos hay que eliminar de la nube de puntos los puntos más alejados y aquellos que invalidarían el mapa (voladizos) para posteriormente poder construirlo.

El esquema implementado en ROS se muestra en la Figura 1, donde se pueden observar los dos nodos creados para la obtención de mapas, *filter* y *map*. Ambos se han incluido en un paquete llamado *pc-processing*.

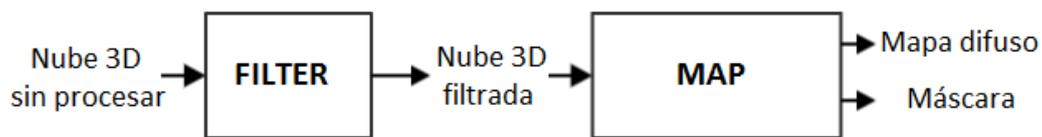


Figura 1: Esquema de nodos en ROS.

#### 3.1. Filtrado

Un método popular de representar el terreno son los mapas  $2\frac{1}{2}$  D o mapas de elevación. Sin embargo, usar nubes de puntos en bruto, que puedan contener voladizos, produciría mapas poco fiables. Por lo tanto, la eliminación de los voladizos atravesables se hace indispensable. Esta tarea se ha abordado calculando la variación de la altura de todos los puntos que se encuentran en cada celda de una cuadrícula 2D y, si existen variaciones de alturas superiores a un determinado umbral  $\sigma$ , relacionado con la altura del robot (0.76 m), se clasifica como voladizo [2].

Identificar y eliminar los voladizos de la nube de puntos se puede considerar un problema de clasificación donde es necesario distinguir entre suelo y voladizo. Esta clasificación implica el procesamiento de una gran cantidad de puntos provenientes de un telémetro láser 3D. Se requieren estructuras de datos y algoritmos eficientes para el procesamiento de datos en 3D. Se ha decidido utilizar Grandes Cubos Binarios (*Coarse Binary Cubes, CBC*) como estructura de datos para implementar la identificación y eliminación de los voladizos en la nube de puntos.

Así, se emplea una cuadrícula 3D compuesta de cubos regulares de arista  $E$  (0.5 m), cuyos valores Booleanos se establecen a uno si cualquiera de los puntos Cartesianos se encuentra dentro de sus límites.

Se asume que el eje  $Z$  de la nube de puntos apunta hacia arriba. Sea  $x_{min}$ ,  $x_{max}$ ,  $y_{min}$ ,  $y_{max}$ ,  $z_{min}$  y  $z_{max}$  las coordenadas cartesianas del cubo mínimo que pueda contener toda la nube de puntos. Entonces, a cada punto Cartesiano de la nube de puntos  $(x, y, z)$  le corresponde el índice  $i$  de su cubo:

$$i = i_x + i_y i_{x_{max}} + i_z i_{x_{max}} i_{y_{max}} \quad (1)$$

donde:

$$i_{x_{max}} = \text{floor} \left( \frac{x_{max} - x_{min}}{E} \right) + 1 \quad (2)$$

$$i_{y_{max}} = \text{floor} \left( \frac{y_{max} - y_{min}}{E} \right) + 1 \quad (3)$$

e  $i_x$ ,  $i_y$ ,  $i_z$  son las coordenadas del cubo que lo contiene:

$$i_x = \text{floor} \left( \frac{x - x_{min}}{E} \right) \quad (4)$$

$$i_y = \text{floor} \left( \frac{y - y_{min}}{E} \right) \quad (5)$$

$$i_z = \text{floor} \left( \frac{z - z_{min}}{E} \right) \quad (6)$$

De esta forma, un punto Cartesiano puede ser definido por las coordenadas  $(i, x, y, z)$ , donde varios puntos pueden tener el mismo índice  $i$ , significando esto que están contenidos por el mismo cubo. Cada índice  $i$  está unívocamente relacionado con las coordenadas del cubo  $(i_x, i_y, i_z)$  de la forma:

$$i_x = \text{remainder} \left( \frac{i}{i_{x_{max}}} \right) \quad (7)$$

$$i_y = \text{floor} \left( \text{remainder} \left( \frac{i}{i_{x_{max}} * i_{y_{max}}} \right) \right) \quad (8)$$

$$i_z = \text{floor} \left( \frac{i}{i_{x_{max}} * i_{y_{max}}} \right) \quad (9)$$

Estas ecuaciones se han simplificado respecto al artículo original [2], ya que en esta implementación se toman como numeros enteros y al realizar la conversion de tipos queda el índice correcto.

*CBC* emplea las siguientes estructuras de datos:

- *Ixyz* es una matriz que contiene las coordenadas  $(i, x, y, z)$  de todos los puntos.
- *L* es una lista de números enteros ordenados de menor a mayor con los índices *i* de cada cubo que contienen algún punto, es decir, es el vector que contiene los índices *i* de cada uno de los puntos 3D sin repeticiones.

Estas estructuras de datos se pueden usar para clasificar los puntos 3D como voladizos o no. El primer paso es filtrar los puntos mediante la distancia de su proyección horizontal al telémetro láser. La razón de esto es, primero, reducir el número de puntos que se van a procesar y, segundo, dado que el telémetro láser 3D está a una altura pequeña sobre el suelo (0.73 m) no tiene una gran visibilidad [3]. Por ello se limita a las zonas más fiables de la nube de puntos 3D, es decir, los alrededores del sensor 3D. En concreto, se ha limitado la zona de confianza a  $d_{max} = 8$  m de radio centrada en el robot.

El proceso completo para clasificar una nube de puntos comienza calculando el índice *i* de cada uno de los puntos de la nube [2] y construyendo la matriz *Ixyz*. Con la primera coordenada de esta matriz (el índice *i*) se construye *L* eliminando las repeticiones y ordenándolo de forma ascendente (utilizando la función *unique()* de la librería *armadillo* [4] para *C++*). Se recorre *L* desde su inicio. Como todas las columnas son procesadas a la misma vez es necesario realizar un seguimiento de la posición del suelo, lo cual se realiza mediante una matriz *S*, que se inicializa con un valor negativo y se actualiza con el índice del suelo en cada una de las columnas de la rejilla.

Al estar *L* ordenada se procesan los cubos de cada columna de la rejilla desde abajo hacia arriba [2]. El primer cubo ocupado se considera como suelo, después, si la distancia entre dicho cubo y el siguiente cubo ocupado es menor que  $\sigma$ , el cubo superior se convierte en suelo. Los puntos contenidos en estos cubos considerados suelo se añaden a la nube de puntos *F* (para ello se ha utilizado la función *find()* de la librería *armadillo*), que será la que contenga los puntos filtrados no eliminados. En el caso de que la distancia supere a  $\sigma$  el cubo superior se clasifica como voladizo. El umbral debe ser mayor que la altura del robot y en el caso de Andábata  $\sigma = 2$ , correspondiendo a 1 m. A diferencia de lo propuesto en [2], aquí se devuelven directamente los puntos no eliminados.

### 3.2. Creación de mapas

Los mapas de elevación ofrecen una forma compacta de representar el terreno, no obstante, tienen limitaciones con la incertidumbre de los sensores o la falta de datos. Por esta razón las herramientas capaces de gestionar estas debilidades, como la lógica difusa, pueden ser utilizadas para modelar el terreno [5] [6]. Como novedad respecto a [6], se ha utilizado identificación borrosa por mínimos cuadrados [7] en lugar de ANFIS. Además, el cálculo de mínimos cuadrados se realiza recursivamente.

El modelado difuso es capaz de hacer frente a los datos del sensor que están distribuidos de manera desigual, son ruidosos e incompletos. Sin embargo, la naturaleza de las reglas de interpolación difusas puede conducir a soluciones poco fiables en áreas con escasez de datos, o cuando no hay lecturas disponibles. En consecuencia, la evaluación de la fiabilidad de las estimaciones difusas es necesaria para el uso del mapa. Para ello se ha construido una máscara de confianza teniendo en cuenta la disponibilidad de puntos en cada una de las regiones. La máscara de confianza se puede superponer al mapa difuso para crear un mapa de elevación proveniente de una nube de puntos con una resolución mínima deseada.

Se emplea una rejilla polar (ver Figura 2) que divide la superficie en sectores y rangos de distancias. Para la identificación borrosa se han utilizado la pertenencia de un punto a cada una de las celdas de la rejilla dependiendo del ángulo y distancia del punto al sensor láser. Para ello, a cada punto 3D se le asignan unas coordenadas cilíndricas a partir de sus coordenadas Cartesianas de la forma:

$$\begin{aligned}\tau &= \sqrt{x^2 + y^2} \\ \alpha &= \arctan\left(\frac{y}{x}\right) \\ z &= z\end{aligned}\tag{10}$$

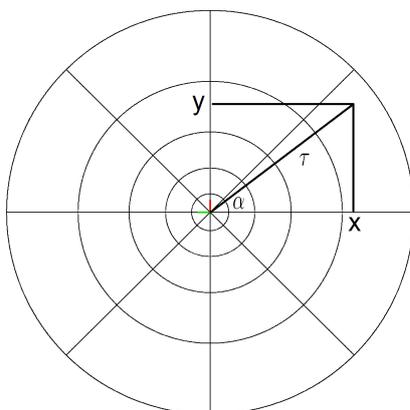


Figura 2: Ejemplo de rejilla con 8 sectores angulares y 5 rangos de distancias.

Usando consecuentes Sugeno de orden cero, cada celda queda definida por un solo parámetro  $a_{ij}$ , que será la altura  $z_{ij}$  de la celda  $(i, j)$  para el mapa borroso y la confianza  $c_{ij}$  en dicha celda para la máscara de confianza.

La pertenencia de un punto a un sector angular  $j$  se calcula:

$$\mu_j(\alpha) = \frac{1 - dif\_ang}{inc\_ang} \quad (11)$$

donde  $dif\_ang$  es el valor absoluto de la diferencia de ángulo entre el punto y el centro de la celda, e  $inc\_ang$  es la diferencia de ángulo entre el centro de una celda cualquiera y la consecutiva (una constante).

La pertenencia de un punto a un rango de distancias  $i$  se calcula como:

$$\mu_i(\tau) = \begin{cases} 1 & \text{si } \tau < \tau c_1 \text{ ó } \tau > \tau c_{conj\_dist} \\ \frac{\tau - \tau c_{i-1}}{\tau c_i - \tau c_{i-1}} & \text{si } \tau > \tau c_1 \text{ y } \tau < \tau c_{conj\_dist} \text{ y } \tau c_i - \tau > 0 \\ \frac{\tau c_{i+1} - \tau}{\tau c_{i+1} - \tau c_i} & \text{si } \tau > \tau c_1 \text{ y } \tau < \tau c_{conj\_dist} \text{ y } \tau c_i - \tau < 0 \end{cases} \quad (12)$$

donde  $\tau c_i$  es la distancia del centro del rango de distancias  $i$  al centro del mapa, y  $conj\_dist$  es el número de rangos de distancia que tiene el mapa (una constante).

La pertenencia de un punto a una celda puede ser calculada usando el operador producto:

$$\mu_{ij}(\tau, \alpha) = \mu_i(\tau) \mu_j(\alpha) \quad (13)$$

La altura de un punto en el mapa borroso ( $z(\tau, \alpha)$ ) y la confianza de un punto en la máscara de confianza ( $c(\tau, \alpha)$ ) puede ser calculado como:

$$z(\tau, \alpha) = \sum_{\forall i,j} \mu_{ij}(\tau, \alpha) z_{ij} \quad (14)$$

$$c(\tau, \alpha) = \sum_{\forall i,j} \mu_{ij}(\tau, \alpha) c_{ij} \quad (15)$$

Como se puede ver en la Figura 3, las funciones de pertenencia de los conjuntos borrosos triangulares son iguales y uniformemente distribuidas para el ángulo, pero, para la distancia, los picos se expanden según están más alejadas del origen de medida.

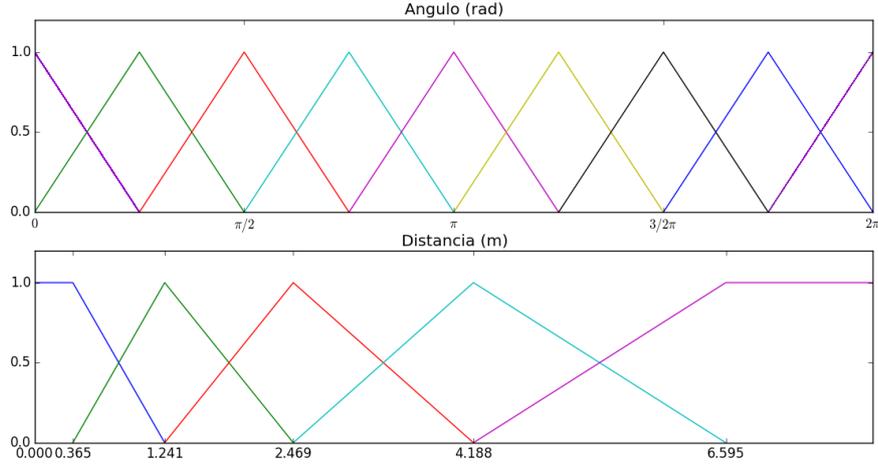


Figura 3: Función de pertenencia en ángulo (8 sectores) y distancia (5 rangos).

Las primera y última funciones de distancia son trapezoidales debido a que a partir de una distancia mínima y máxima la pertenencia del punto es completa. Los picos a uno  $d_i$  de las funciones de pertenencia se calculan mediante la fórmula:

$$d_i = \frac{r^i - 1}{r^k - 1} d_{max} \quad (16)$$

donde  $k$  es el número de conjuntos en distanciam y  $r$  es un parámetro de expansión que se calcula despejándolo de la siguiente ecuación:

$$\frac{r^1 - 1}{r^k - 1} d_{max} = d_1 \quad (17)$$

donde  $d_1$  es el radio del primer rango de distancias, que para el caso de Andábata es de 0.73 m debido a que es la distancia ciega alrededor del robot [3].

La identificación por minimos cuadrados recursivos de las alturas de las celdas se realiza mediante los siguientes calculos recursivos:

$$L_1 = \frac{P_1 m}{1 + m' P_1 m} \quad (18)$$

$$H = H + L (z - m H) \quad (19)$$

$$P_1 = P_1 - L_1 m' P_1 \quad (20)$$

donde  $H$  es el vector con las alturas  $z_{ij}$  de cada una de las celdas,  $m$  es el vector de pertenencia del punto a cada una de las celdas de la rejilla,  $z$  es la altura del punto, y  $P_1$  es la matriz de covarianza que se escoge inicialmente con un valor de  $10 * \text{matriz\_identidad}$  (se presupone que el terreno va a ser plano alrededor del robot).

Una vez calculada  $H$  la altura de un punto del plano se calcula como:

$$z = m H \quad (21)$$

Para el cálculo de la confianza las ecuaciones recursivas son:

$$L_2 = \frac{P_2 m}{1 + m' P_2 m} \quad (22)$$

$$M = M + L (1 - m M) \quad (23)$$

$$P_2 = P_2 - L_2 m' P_2 \quad (24)$$

donde  $M$  es la máscara con las confianzas  $c_{ij}$  de cada una de las celdas, la cual se inicializa con una confianza de 0 para cada una de las celdas, y  $P_2$  es la matriz de covarianza que se escoge inicialmente con un valor de  $0,1 * \text{matriz\_identidad}$  (se presupone que la confianza inicial para todas las celdas es muy baja).

Y la confianza en un punto cualquiera del plano se calcula como:

$$c = m M \quad (25)$$

En la zona central de la nube de puntos, dado que no hay puntos disponibles,  $H$  se inicializa creando un plano perpendicular al eje  $Z$  del robot y que contiene el origen de coordenadas del robot, después se evalúan las coordenadas de los centros de las celdas para que pertenezcan a dicho plano calculando su altura. Para el caso de la máscara de confianza la zona central se fija a 1, dando por válida la altura inicial y la inclinación del robot.

Para aumentar la velocidad del proceso y dado que el procesador de Andábata cuenta con 8 núcleos [8], se paraleliza el cálculo del mapa y de la máscara por mínimos cuadrados, como muestra la Figura 4. Como se puede apreciar, tanto en el proceso del calculo de las alturas como en el de la máscara de confianza se ha de calcular el vector  $m$  de pertenencia del punto a cada celda de la rejilla. Debido a la paralelización de ambos procesos este cálculo se ha de hacer por repetido. Esto no es un inconveniente, ya que el beneficio de la paralelización es mucho mayor que el perjuicio de duplicar dicho cálculo.

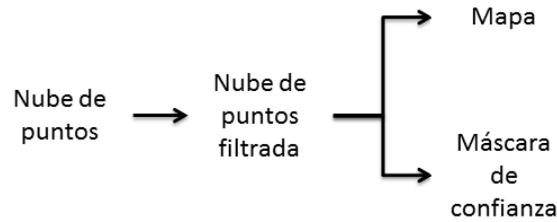


Figura 4: Diagrama del flujo de datos durante la construcción del mapa.

## 4. Resultados experimentales

Para este apartado se han tomado una serie de barridos 3D estáticos desde Andábata en un parque urbano con árboles alrededor y algunos obstáculos más (ver Figura 5). Se ha escogido una nube de puntos de las capturadas y se ha procesado con los métodos descritos en este informe. Se pretende comprobar si la implementación de los métodos es correcta y los tiempos de procesado.



Figura 5: Entorno en el que se han realizado los experimentos.

La nube de puntos 3D capturada tiene 148614 puntos (ver Figura 6a). Como se puede ver en la Figura 6b, el filtro elimina satisfactoriamente los elementos voladizos, como pueden ser las copas de los árboles, y deja intactos el resto de elementos a una distancia inferior a 8 m sobre el plano  $XY$ . A la salida del filtro quedan 107960 puntos, tras un tiempo de procesado de  $0.7 \pm 0.05$  segundos en la computadora de Andábata [8]. Los puntos que no han sido eliminados por el filtro son los que se utilizan para crear el mapa.

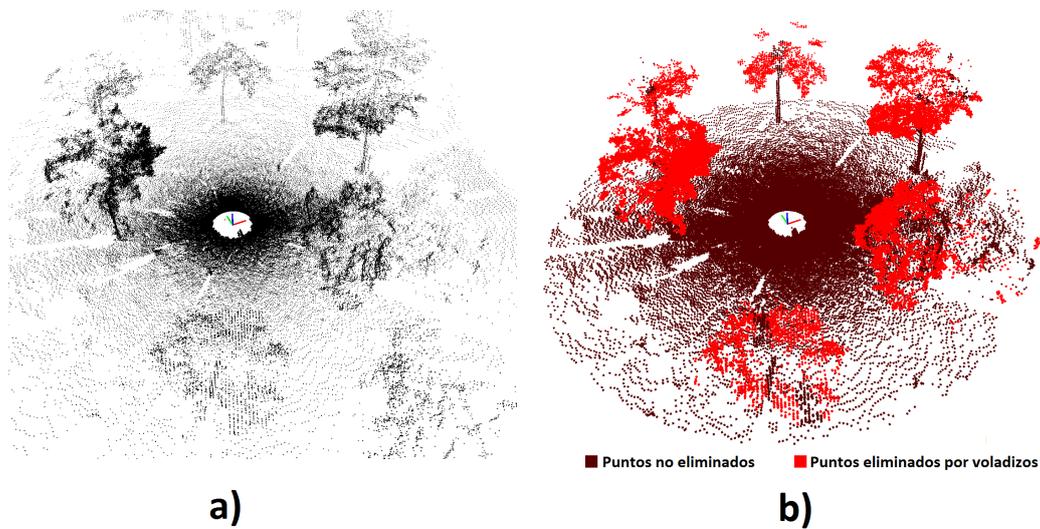


Figura 6: Vista del barrido 3D a) y la salida del filtro tras procesarlo b).

Se ha utilizado una rejilla de 16 sectores angulares y 12 rangos de distancia. El procesado de los puntos para crear el mapa tomó en un primer momento  $56 \pm 5$  segundos, este tiempo se mejoró hasta un valor de  $31 \pm 5$  segundos gracias a la paralelización del cálculo de mínimos cuadrados, como se comentó en la sección 3.2. Se ha comprobado que el tiempo de procesado depende del número de puntos, por lo que se puede reducir si la nube de puntos se captura con menor resolución, por ejemplo.

Como se observa en la Figura 7a las elevaciones del mapa coinciden con los obstáculos reales haciendo posible la planificación con dicho mapa. En este caso concreto, dado que la densidad de puntos es elevada y no hay nada demasiado grueso que obstruya la visión, la máscara ha finalizado con gran confianza en todas las regiones, como se puede ver en la Figura 7b, donde todas las celdas tienen una confianza cercana a 1.

El mapa creado con lógica difusa tiene la suficiente resolución para planificar la navegación local, gracias a la máscara de validación, e identifica los obstáculos y la topología del terreno adecuadamente.

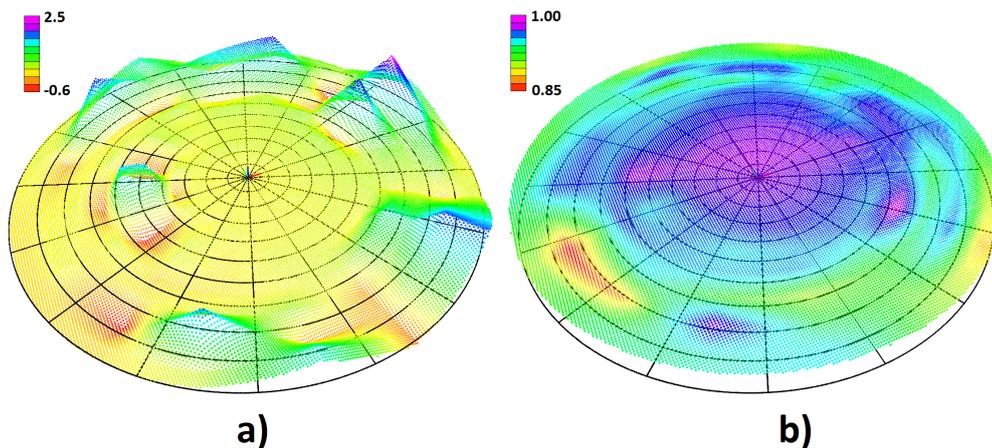


Figura 7: Mapa de elevación borroso a) y mascara de confianza en cada celda b).

## 5. Referencias

- [1] M. Zafra, J. L. Martínez, y J. Morales, “Obtención de Barridos 3D Nivelados mientras Andábata se Desplaza,” *Inf. Tec. Universidad de Málaga*, Sep. 2015.
- [2] A. J. Reina, J. L. Martínez, A. Mandow, J. Morales, y A. García-Cerezo, “Collapsible Cubes: Removing Overhangs from 3D Point Clouds to Build Local Navigable Elevation Maps,” en *Proc. IEEE/ASME International Conference in Advanced Intelligent Mechatronics*, (Besanson, Francia), pag. 1012–1017, 2014.
- [3] J. L. Martínez, J. Morales, A. J. Reina, A. Mandow, A. Pequeño-Boyer, y A. García-Cerezo, “Construction and Calibration of a Low-cost 3D Laser Scanner with Full Field-of-view for Mobile Robots,” en *Proc. IEEE International Conference on Industrial Technology*, (Sevilla, España), pag. 149–154, 2015.
- [4] C. Sanderson, “Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments ,” *Inf. Tec. NICTA*, 2010.
- [5] A. Mandow, T. J. Cantador, A. García-Cerezo, A. J. Reina, J. L. Martínez, y J. Morales, “Fuzzy Modeling of Natural Terrain Elevation from 3D Scanner Point Cloud,” en *Proc. IEEE 7th International Symposium on Intelligent Signal Processing(WISP)*, (Floriana, Malta), pag. 1–5, 2011.
- [6] J. Martinez, A. Mandow, A. Reina, T. Cantador, J. Morales, y A. Garcia-Cerezo, “Navigability Analysis of Natural Terrains with Fuzzy Elevation Maps from Ground-based 3D Range Scans,” en *Proc. IEEE/RSJ International Conference in Intelligent Robots and Systems*, (Tokio, Japón), pag. 1576–1581, 2013.

- [7] A. García-Cerezo, M. López-Baldán, y A. Mandow, “An Efficient Least-Squares Fuzzy Modelling Method for Dynamic Systems,” en *Proc. IMACS Multiconference, Symposium on Modelling, Analysis and Simulation (CESA)*, (Lille, Francia), pag. 885–890, 1996.
- [8] M. Zafra, “Construcción de Mapas de Exteriores Mediante Octrees para un Robot Móvil equipado con un Telémetro Láser 3D,” Proyecto Final de Carrera (E.T.S.I. Industrial), Universidad de Málaga, 2015.