

# Construcción de mapas mediante características visuales para aplicaciones en robótica de servicio

Karen Lizbeth Flores-Rodriguez<sup>1</sup>,  
Felipe Trujillo-Romero<sup>2</sup>, José-Joel González-Barbosa<sup>1</sup>

<sup>1</sup> Instituto Politécnico Nacional, CICATA-Querétaro,  
México

<sup>2</sup> Universidad de Guanajuato, Departamento de Ingeniería Electrónica, DICIS-UG,  
México

karenflores350@hotmail.com, jgonzalezba@ipn.mx fdj.trujillo@ugto.mx

**Resumen.** En este trabajo se presenta el desarrollo de un algoritmo para la construcción de mapas bidimensionales mediante odometría inercial y elementos visuales. Se hace uso de un módulo de reconocimiento de objetos basado en características locales y redes neuronales artificiales no supervisadas. El módulo se utiliza para aprender los elementos no dinámicos en una habitación y asociarles una posición. El mapa queda representado como una red neuronal a la cual cada neurona le corresponde una posición real. Los experimentos se realizaron mediante simulación en Webots y con un robot NAO virtual. Una vez construido el mapa sólo basta con capturar un par de imágenes del entorno para estimar la ubicación del robot. Los resultados demuestran que los mapas bidimensionales alcanzan una precisión de hasta  $\pm(0,06, 0,1)$  m.

**Palabras clave:** elementos visuales, mapas bidimensionales, odometría inercial, robot humanoide NAO, A-KAZE, GCS.

## Maps Construction Using Visual Features to Service Robotics Applications

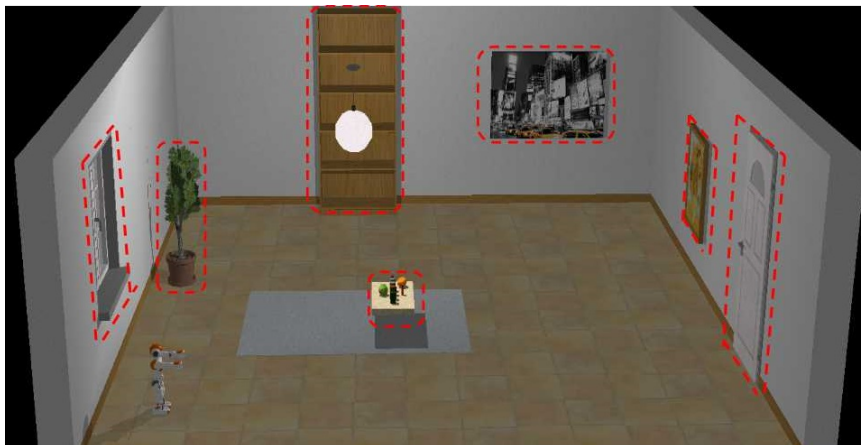
**Abstract.** This paper presents a map construction algorithm development by inertial odometry and visual features. It uses an object recognition module based on local features and unsupervised artificial neural networks to learn no dynamic elements in a room and assign them a position. The map represent a neural network where each neuron is a real position in the room. The experiments were made by simulation in Webots environment using the virtual humanoid robot NAO. Once the map is built, it is enough to capture a couple of images from the environment to estimate the location of the robot. The results show a good precision in localization with the two dimensional maps through  $\pm(0,06, 0,1)$  m.

**Keywords:** visual features, bidimensional maps, inertial odometry, humanoid robot NAO, A-KAZE, GCS.

## 1. Introducción

Crear a un ser artificial ha sido el sueño del hombre desde que nació la ciencia. Ya que siempre ha tratado de imitar, igualar y mecanizar la inteligencia humana en máquinas que puedan ejecutar tareas para cumplir un propósito. El desarrollo de robots humanoides es una prueba de la creación de una entidad que puede pensar y servir, considerados robots de servicio. Según la Federación Internacional de Robótica (IFR) [4], un robot de servicio es un robot que opera de forma parcial o totalmente autónoma, para realizar servicios útiles para el bienestar de los humanos y del equipamiento, excluyendo operaciones de manufactura. Entendiendo como servicio a una actividad que se realiza para el beneficio de otros. Los robots de servicios pensados para hogares, hospitales, restaurantes, etc. deben tomar decisiones complejas, como identificar el medio con el que interactúan, detectar su objetivo (objetos) y cumplir órdenes (reconocimiento, manipulación) de manera autónoma.

Para que un robot de servicio sea autónomo es necesario que cuente con un sistema de control que le permita interactuar con el medio en el que se encuentra para tomar decisiones correctas y cumplir metas concretas. Una pieza importante del sistema de control de los robots de servicio es el aprendizaje del entorno donde trabajarán. Este tipo de robots, primero debe conocer el lugar y los elementos no dinámicos con los que va a interactuar. Por ejemplo, en la competencia RoboCup@Home [5], los competidores cuentan con un día y medio para conocer los escenarios en los que van a interactuar y realizar las calibraciones necesarias para el cumplimiento de las tareas propuestas.



**Fig. 1.** El marco de referencia global es el punto del cual el robot inicia un recorrido en circuito de la habitación. Las imágenes que capture alrededor le permiten conocer los elementos no dinámicos del ambiente.

El aprendizaje del entorno presentado en este trabajo le permite a un robot construir un mapa bidimensional del lugar donde se encuentra para estimar su ubicación con respecto a un marco de referencia global del mapa construido. Para la construcción del mapa bidimensional se utiliza la pose (posición más orientación) del robot combinado con elementos visuales. Por ejemplo, en la figura 1, el robot comenzará un recorrido de circuito cerrado alrededor de la habitación de la cual construye el mapa. Éste considera el marco de referencia global como el punto del cual partió. Va capturando imágenes de la pared más cercana a cada paso que dé. Las imágenes capturadas las asociará a la posición de donde las tomó para realizar la construcción del mapa bidimensional que incluye los elementos no dinámicos del ambiente donde trabajará. Una vez aprendido el mapa, el uso de éste se basa en la detección de los elementos no dinámicos del ambiente para estimar su posición con respecto al marco de referencia sin importar si ha perdido su ubicación odométrica.

### **1.1. Estado del arte**

Los trabajos sobre construcción de mapas para robots de servicio en ambientes humanos son comunes hoy en día. De hecho, competencias como RoboCup@home tienen como objetivo el desarrollo de tecnología robótica de servicio y asistencia [5]. La ventaja de esta competencia es el uso de robots desarrollados por los equipos participantes con diferente tipo de sensores y actuadores. Este tipo de robots cuentan con sensores de profundidad además de cámaras para la captura de información. Los sensores permiten obtener información de distancia e imagen para obtener mapas más precisos como en [6-9]. Usar solamente información proveniente de cámaras es suficiente y disminuye costos cuando se realiza un buen procesamiento de imágenes. Para la construcción de mapas en interiores con el uso de una sola cámara es muy utilizado localización y mapeado simultáneo (SLAM) visual, trabajos donde se utiliza este método y la cámara de un robot se pueden encontrar en [10-14]. En este trabajo se utiliza al robot humanoide NAO [3] como plataforma para la implementación. La adquisición de imágenes se realiza mediante una de sus cámaras y se utiliza su unidad de medición inercial para obtener los datos odométricos. Algunos trabajos que utilizan al robot NAO para la construcción de mapas son [15-18]. El aprendizaje del entorno se realiza haciendo uso de un módulo de reconocimiento de objetos presentado en [1], en el cual se plasman diversos experimentos que demostraron su excelente ejecución. Este módulo utiliza el descriptor de características A-KAZE y la red neuronal auto-organizada Growing Cell Structure (GCS) para aprender y reconocer objetos. A-KAZE [19] se basa en KAZE [20], su mejora recae en que es más rápido gracias al incremento en velocidad conseguido por el esquema Fast Explicit Diffusion (FED).

Además, muestra una demanda computacional y requerimiento de almacenamiento menor gracias al descriptor invariante a rotación y escala Modified-Local Difference Binary (M-LDB). Este método cuenta con mejor ejecución que SURF, SIFT, KAZE, ORB and BRISK. El módulo usa la variante de la red neuronal no supervisada de Kohonen la red GCS [21]. La principal ventaja de la red

es su habilidad para ajustarse automáticamente a cierta estructura y tamaño basada en los datos de entrada, alcanzada gracias a un proceso de crecimiento controlado con ocasionales remosiones de neuronas. El model utiliza estructuras de hiper-tetraedros debido a su mínima complejidad y su combinación fácil en grandes estructuras. Las contribuciones de la construcción de un mapa bidimensional por parte de un robot de servicio presentado en este trabajo son: (i) la captura de información de manera autónoma en los recorridos en circuito cerrado de la habitación, (ii) la combinación de elementos visuales y posicionamiento odométrico, (iii) el aprendizaje de la posición de los elementos no dinámicos utilizando redes neuronales no supervisada, (iv) la obtención de mapas bidimensionales como redes neuronales en donde cada neurona tiene una posición real, (v) la fácil estimación de la ubicación del robot en el mapa mediante el reconocimiento de los objetos no dinámicos.

## 2. Construcción de mapa bidimensional

Para este trabajo se utilizan las mediciones incrementales de los encoders en las articulaciones y la cámara del propio robot humanoide NAO[3] para ir almacenando su ubicación con respecto a un marco de referencia global. Ambas herramientas le permiten al robot la construcción de un mapa bidimensional del lugar en el que está navegando. La construcción del mapa bidimensional se lleva a cabo mediante dos herramientas: 1) Odometría y 2) Elementos Visuales. Ambos enfoques van a permitir construir un mapa bidimensional en el plano  $XY$  del entorno donde el robot navegue.

### 2.1. Odometría

La odometría permite estimar la posición relativa de un robot o vehículo en el plano durante la navegación desde su localización inicial. En este trabajo se utiliza para determinar y guardar la ubicación del robot durante su recorrido en la habitación para la construcción del mapa bidimensional. El robot NAO cuenta con funciones que apoyan en la resolución de varios problemas, uno de ellos es la odometría. Por razones prácticas se decide utilizar estas funciones. En el Algoritmo 1 se observa el pseudo código donde se hace uso de las funciones de odometría inercial de Aldebaran [3]. En esta implementación se inicializa la posición bidimensional del robot mediante la función  $pose2D(X, Y, theta)$  con valores explícitos recuperados con la función  $getRobotPosition()$ . Los valores con los que se inicializa la pose se recuperan desde los encoders magnéticos rotatorios (MRE) de las articulaciones. Éstos están contenidos en un vector con la posición absoluta del robot en el mundo  $(X, Y)_m$  y un ángulo  $theta$   $Wz$  en radianes  $(-\pi, \pi)$ . Cada que el robot se enciende almacena una posición absoluta en el mundo. En la construcción del mapa bidimensional, después de guardar la posición inicial, se le indica al robot que realice el recorrido de circuito cerrado en la habitación avanzando cierta distancia al caminar. Al avanzar cierta distancia se vuelve a almacenar la posición bidimensional. Posteriormente se

---

**Algoritmo 1:** Pseudo código para almacenar odometría inercial basado en funciones de Aldebaran [3].

---

```
1 //Almacenamiento de la posición inicial
2 AL::Math::Pose2D worldToRobotInit= Pose2D(getRobotPosition())
3 //Espera hasta que termine de desplazarse
4 //Almacenamiento de la posición final
5 AL::Math::Pose2D worldToRobotAfter= Pose2D(getRobotPosition())
6 Pose2D robotMove = pose2DInverse(worldToRobotInit)+worldToRobotAfter
7 //Desplazamiento
8 theta = modulo2PI(robotMove.theta)//Ángulo
```

---

calcula el desplazamiento que realizó el robot y el ángulo. Utilizando la función *pose2DInverse* se calcula la posición bidimensional entre la actual y la siguiente. El ángulo se obtiene mediante *modulo2PI* el cual regresa un ángulo entre  $-\pi$  y  $\pi$ .

Considerando un panorama más amplio, se establece el siguiente algoritmo generalizado:

1. Captura de posición del robot con respecto al mundo antes de comenzar a caminar.
2. Detección de inicio del caminado del robot.
3. Simultáneamente, inicio de la recolección de datos odométricos.
4. Procesamiento y acumulación de datos odométricos.
5. Detección de finalización del caminado del robot. Si no es así se repiten pasos 3 y 4.
6. Cálculo de la distancia que recorre el robot.
7. Almacenamiento de la distancia y posición del robot para la construcción del mapa bidimensional.

## 2.2. Elementos visuales

El uso de elementos visuales se basa en analizar y crear una base de detalles existentes en el entorno tomando en cuenta la posición del robot en la cual se captura la imagen. Esta base de datos se crea de manera similar a la base de datos de objetos en el módulo de reconocimiento de objetos [1]. En lugar de diferentes imágenes del objeto a reconocer como entrada del módulo, aquí son necesarias diferentes vistas de la habitación. Se considera que los elementos visuales más significativos para poder realizar una representación del entorno son los que se encuentran en las paredes o cerca a ellas.

Con los elementos visuales y la ubicación estimada mediante odometría se construye el mapa bidimensional de la habitación que recorra el robot. Por ejemplo, en la figura 2, se muestra una habitación simulada mediante el software Webots [2]. En esta habitación se observan diversos objetos que podrían encontrarse en cualquier casa: cuadros, mesas, estantes, plantas, ventanas, puertas, etc. Estos objetos tienen una posición determinada y es sabido que no se moverán de

su lugar. El robot tiene que recorrer esta habitación en un circuito cerrado, de preferencia cuadrangular, tomar imágenes y almacenar la posición estimada de donde se ha tomado la captura. El robot debe enfocar la captura de imágenes a



Fig. 2. Simulación de una habitación en Webots.

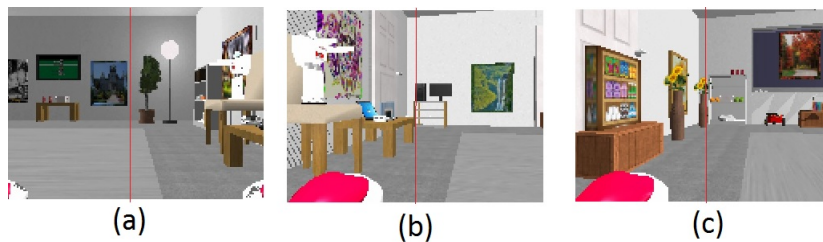


Fig. 3. Capturas realizadas por el robot NAO en habitación simulada en Webots. Las capturas se almacenan junto con la pose de donde las tomó. (1) Pose:(4,3), (2) Pose: (0,3), (3) Pose: (3,0) en metros.

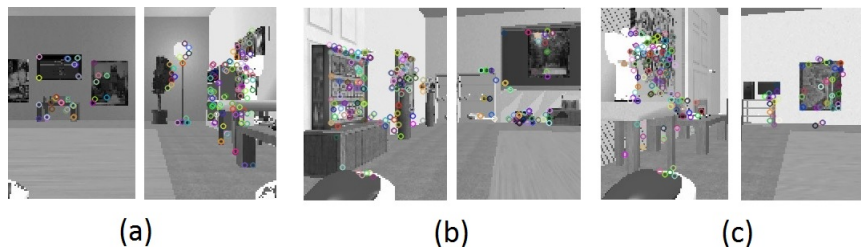
la pared más cercana por donde él vaya caminando. En la figura 3, se observan tres capturas realizadas por el robot en diferentes puntos. Mientras realiza el recorrido, tomará una captura cada ciertos pasos dependiendo el número de imágenes que el usuario indique capturar de la habitación. Por ejemplo, si se requieren 20 imágenes en una habitación de 4 metros por pared, se tomará 1 imagen cada 20 cm. Además del número de capturas y dimensión de la pared de

la habitación, también se puede decidir cuantas veces se realizará el recorrido. Entre más recorridos se realicen mejor se construirá la habitación. Una vez terminado el o los recorridos, con la información almacenada por parte del robot se procede a la construcción del mapa bidimensional. Las imágenes capturadas contendrán objetos o partes de objetos de los cuales se tiene que adquirir cierta información. Se utiliza el módulo de reconocimiento de objetos, descrito en [1], para la extracción de esta información. Utilizando este módulo se obtienen los descriptores de la imagen para aprender la información y asociarla con la pose del robot al momento de capturarla.

Esta información se fusiona en una representación bidimensional la cual será el mapa de la habitación. Antes de iniciar cualquier recorrido por la habitación, el robot debe detectar la pared más cercana para saber hacia donde tiene que girar su cabeza. Mientras realiza su recorrido irá tomando imágenes sólo girando su cabeza hacia la pared detectada. Al momento de que el robot deba girar para recorrer la siguiente pared, basta con girar en la dirección opuesta a donde detectó la pared para continuar con el recorrido en la habitación. La detección de la pared se lleva a cabo visualmente. Antes de comenzar a caminar, el robot debe ser colocado de manera paralela a cualquier pared y en una esquina de la habitación.



**Fig. 4.** Imágenes tomadas por robot NAO antes de iniciar recorrido en la habitación para detectar la pared divididas en dos, pared más cercana: (a) lado derecho, (b) lado izquierdo, (c) lado izquierdo.



**Fig. 5.** Detección de pared mediante descriptores.

Entonces, éste capturaré una imagen viendo hacia al frente para analizarla. La imagen se analiza dividiéndola en dos, por ejemplo, en la figura 4, se observan tres capturas de la habitación en diferentes posiciones. En la imagen (a) la pared más cercana se encuentra a la izquierda, mientras que en las imágenes (b) y (c) están a la derecha. Cada una de las imágenes se dividen en dos de manera vertical y se obtienen los puntos característicos o puntos salientes de cada uno de los lados. Estos puntos salientes se obtienen con el método A-KAZE utilizado en el módulo de reconocimiento de objetos.

En la imagen que obtenga más puntos salientes es donde se encontrará la pared, partiendo de la restricción de que la habitación se encuentra libre de obstáculos. En la figura 5 se muestran los resultados de la evaluación de cada imagen. En éstas se observan los puntos salientes señalados con pequeños círculos de colores. En (a) la imagen con más puntos salientes es la derecha con 108 contra 36, en (b) la izquierda con 128 contra 50 y en (c) la izquierda con 119 contra 53. Entonces el robot gira su cabeza hacia esa dirección para ir aprendiendo la habitación.

### 2.3. Algoritmo para la construcción del mapa bidimensional

La construcción del mapa bidimensional se lleva a cabo de la siguiente manera. Primero, el Algoritmo 2, tiene la tarea de ejecutar un recorrido en circuito cerrado alrededor de una habitación cuadrada por parte del robot. Éste realiza el circuito en la habitación capturando y guardando imágenes con su respectiva poses. Es necesario conocer,  $d$ , la dimensión de una pared a recorrer de la habitación y  $p$  el tamaño de paso al caminar. Otro parámetro que se puede elegir es cuantas veces realizará el recorrido  $n$ . Si se realizan más de un recorrido el robot reforzará el aprendizaje de la habitación.

Al inicio del algoritmo, el robot realiza una primer captura, *TakePicture()*, para detectar la pared más cercana *DetectNearestWall()*. De esta manera, se obtiene el ángulo, *AngleYaw*, al cual girará mientras realiza su recorrido.

Antes de comenzar a caminar, el robot guarda su pose actual mediante odometría *CurrentPose()*, como punto de referencia global de la habitación  $O_w$ . Entonces se comienza el ciclo de trabajo por número de recorridos  $n$ . Después, se inicializa la variable, *TotalDistanceWalked*, que será el indicador de cuanto lleva recorrido el robot de la distancia total que debe recorrer en toda la habitación. Esta variable se verifica mediante un ciclo de trabajo que no se detendrá mientras la variable no sea igual a  $d \times 4$  (distancia de una pared por las cuatro que conforman la habitación).

Dentro de este ciclo de trabajo se encuentra otro ciclo de trabajo que verifica la variable *DistanceWalked*, (distancia que debe recorrer por pared). Este ciclo de trabajo no se detendrá mientras la variable sea igual a la distancia a recorrer,  $d$ , por pared de la habitación. Dentro de este ciclo de trabajo se ejecuta la odometría inercial y la captura de imagen ( $\mathbf{P}$ ,  $\mathbf{I}$ ), cada que el robot camina  $p$  distancia por paso. Una vez terminado el ciclo, el robot gira su cuerpo, *TurnBody(-AngleYaw)*, hacia el ángulo opuesto al cual giró su cabeza para seguir recorriendo la habitación.



---

**Algoritmo 2:** Módulo de Navegación. *Ejecución de un circuito cerrado.*

---

**Datos:**  $d$  dimensiones de la habitación,  $p$  tamaño de paso del robot al caminar,  
 $n$  número de recorridos

**Resultado:** *data* imágenes y poses

```

1 picture =TakePicture()
2 AngleYaw=DetectNearestWall(picture)
3 TurnHead(AngleYaw)
4  $O_w$ =CurrentPose()
5 TotalDistanceWalked=0
6 per  $j = 1$  a  $n$  fai
7   mientras  $TotalDistanceWalked \neq d \times 4$  hacer
8     DistanceWalked=0
9     mientras  $DistanceWalked \neq d$  hacer
10      Walk( $p$ )
11      RP=CurrentPose()
12      picture =TakePicture()
13       $data = (\mathbf{P}, \mathbf{I})$ 
14      DistanceWalked=DistanceWalked+ $p$ 
15    TurnBody(-AngleYaw)
16    TotalDistanceWalked=TotalDistanceWalked+DistanceWalked

```

---

Después de realizado el circuito y almacenada la base de datos de la habitación, se ejecuta el Algoritmo 3. En este algoritmo, se hace uso del módulo de reconocimiento de objetos para aprender una nueva base de datos con la información de las capturas y las poses  $(\mathbf{P}, \mathbf{I})$ . En éste se le extraen todos los puntos salientes, se construyen histogramas por imágenes y se entrena una red neuronal con GCS. De la red se extraen las clases por imagen obtenidas,  $classes(\mathbf{N})$ . Una vez obtenidas las clases, se verifica si una clase formada por varias imágenes tiene mas de una pose asociada. Si es así se lleva a cabo un promedio entre ellas y se almacena. Posteriormente se utiliza el Algoritmo 4 para utilizar el mapa. El módulo recibe una o más imágenes, de éstas se extraen los puntos salientes y se construyen los histogramas de las imágenes. Los histogramas se envían a evaluar con la red neuronal entrenada para obtener las neuronas a las que pertenece. Una vez conocidas las clases se obtienen las poses a las cuales corresponde y se regresa la posición bidimensional en el mapa. El algoritmo cuenta con ciertas restricciones como conocer la dimensión de la pared de la habitación. De esta manera se calcula la distancia total que va a recorrer el robot alrededor de la habitación. En la habitación no deben de haber obstáculos pues en este trabajo no se está atacando el problema de evasión de obstáculos. Otra restricción es que si algunos elementos en la habitación se han movido de lugar el robot debe volver a construir su mapa de navegación.

---

**Algoritmo 3:** Módulo de Navegación. *Construcción de mapa bidimensional*

---

**Datos:**  $I$  imágenes,  $P$  poses.  
**Resultado:**  $classes(N,P)$  clases del mapa bidimensional.

```
1 per  $i \leftarrow 1$  to  $I$  fai
2   | keypoints =AKAZE( $I(i)$ )
3   |  $H(i)$ =BuildHistos(keypoint)
4 ANN_trained=GCS( $H$ )
5  $classes(N)$ =ANN_trained
6 per  $i \leftarrow 1$  to  $N$  fai
7   |  $NewPose(i) = \frac{1}{n} \sum_{j=1}^n P(classes(i))$ 
```

---

---

**Algoritmo 4:** Módulo de Navegación. *Utilización del mapa bidimensional*

---

**Datos:**  $I$  imágenes  
**Resultado:**  $class(I)$  clases de objetos,  $Pose$  pose

```
1 keypoints =AKAZE( $I$ )
2  $H(I)$ =BuildHistos(keypoint)
3  $classes(I)$ =ANN_trained
4  $Pose = \frac{1}{n} \sum_{j=1}^n NewPose(I)(classes(i))$ 
```

---

### 3. Experimentos y resultados

Los experimentos de esta sección se dividen en dos partes: (1) construcción de mapa bidimensional y (2) ubicación en el mapa bidimensional. En un ambiente semiestructurado libre de obstáculos, donde las dimensiones de una habitación son conocidas, un robot debe navegar para construir un mapa bidimensional del entorno. Este mapa ayuda al robot a conocer en que parte de la habitación se encuentra para realizar posteriores tareas de servicio o regresar al punto del cual partió. Los experimentos se han realizado mediante el simulador Webots y el robot virtual NAO.

#### 3.1. Construcción del mapa bidimensional

Se construyó la habitación simulada en Webots mostrada en la figura 2, de  $6 \times 6$  metros. En esta habitación se colocaron diversos objetos: sillas, mesas, retratos, etc. Se utilizó un robot NAO virtual para la construcción del mapa bidimensional de esa habitación. El robot inició su recorrido desde la esquina inferior izquierda de la imagen quedando ésta como posición global 0 de la habitación. El recorrido lo realizó en circuito cerrado cuadrangular girando su cabeza hacia la pared para poder capturar las imágenes mientras avanzaba. Para mostrar los resultados de la construcción del mapa bidimensional de una manera sencilla, las paredes se han enumerado de 1 al 4 en contra de las manecillas del reloj. El robot realizó dos circuitos cerrados de  $4 \times 4$  alrededor de la habitación

**Tabla 1.** Parámetros principales de los experimentos: Construcción del mapa bidimensional.

Circuito	Imágenes	Pared1	Pared2	Pared3	Pared4
1	89	20	21	31	17
2	75	18	20	24	13

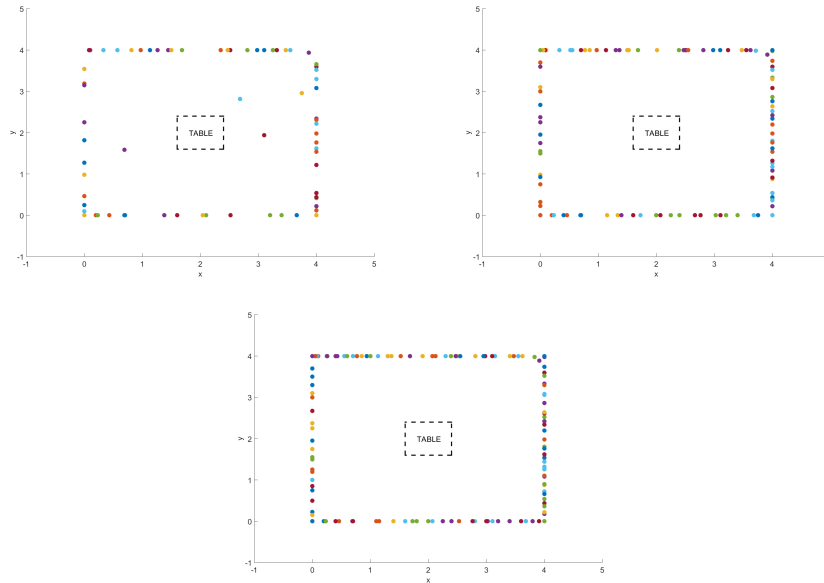
en contra de las manecillas del reloj tomando imágenes y guardando su relación espacial. El número de capturas que realizó por pared se plasman en la Tabla 1. Circuito indica el número de circuito que realizó. Imágenes indica el número total de imágenes que almacenó en el circuito. Pared1, Pared2, Pared3 y Pared 4 indican el número de imágenes correspondientes que almacenó de cada pared. Se almacenaron un total de 164 imágenes y poses utilizada para la construcción del mapa bidimensional. Ya que se utiliza el módulo de reconocimiento de ob-

**Tabla 2.** Parámetros del módulo de reconocimiento de objetos para la construcción del mapa bidimensional.

Experimento	Entrenamiento	Neuronas	Epocas	Tiempo (seg)
1	164	100	100	4.063
2	164	200	200	14.287
3	164	300	300	33.347

jetos, los parámetros correspondientes a los experimentos para este módulo se presentan en la Tabla 2. Experimento indica el número de mapa construido. Entrenamiento indica el número de imágenes utilizadas como entrada en el módulo de reconocimiento de objetos. Además, se indica el número de neuronas y número de épocas seleccionado para el entrenamiento de la red neuronal. Se realizaron tres construcciones de mapa bidimensional, con 100, 200 y 300 neuronas. La idea es observar el desempeño del módulo para la construcción de un mapa bidimensional relacionando lo que el robot observó en su recorrido por la habitación. Los tiempos en segundos obtenidos de los entrenamientos se indican en la misma tabla, se observa que son relativamente pequeños ya que permanecen por debajo del minuto.

Después del entrenamiento se obtuvieron mapas bidimensionales con 72, 111 y 132 poses, en las figura de 6 se presentan los mapas construidos. Para verificar que tan buena fue la construcción del mapa basta con observar que las neuronas quedaron bien distribuidas a lo largo de éste. Cada neurona tiene una pose asociada las cuales son los puntos observados en cada uno de los mapas, es fácil deducir que mientras se incrementa el número de neuronas la distribución de las poses mejora. Es importante mencionar que las poses fueron homogeneizadas en la coordenada que permanecía constante al realizar el recorrido para mostrar una distribución limpia. En el primer mapa construido



**Fig. 6.** Mapa bidimensional experimento 1 (arriba izquierda), 2 (arriba derecha) y 3 (abajo centro), distribución de las neuronas por pose en la habitación.

con 100 neuronas la distribución no es muy buena ya que se observan algunas poses dentro del área donde el robot no ha recorrido, además, en algunas partes hay amontonamiento de éstas. En el mapa construido con 200 neuronas la distribución mejora considerablemente en comparación con el mapa anterior. Sin embargo, en este mapa aun se observan algunos amontonamientos de poses. El mapa construido con 300 neuronas tiene una mejor distribución, se cubren más espacios y aunque aun se observan algunas poses sobre otras, estas son mínimas. Gracias al mapa bidimensional el robot conocerá donde están las paredes para no ir hacia ellas mientras realiza sus tareas. El mapa también apoya al robot al momento de regresar al punto del cual partió considerado la posición global.

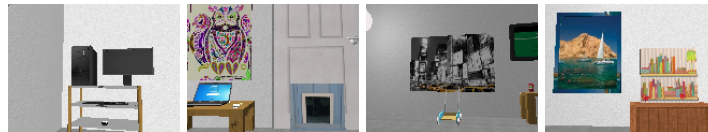
### 3.2. Ubicación en el mapa bidimensional

La idea del uso del mapa bidimensional es, que además de ubicarse en todo momento, una vez que el robot haya terminado sus tareas, éste podrá regresar a la posición global 0 del mapa. Con el mapa construido el robot es capaz de ubicarse en la habitación con una o dos imágenes de las paredes más cercanas. Se realizaron 4 experimentos, en la Tabla 3 se encuentran los parámetros que incluyen número de experimento, mapa bidimensional construido en la sección anterior (1, 2 y 3) y la posición real que se pretende calcular  $(x, y)$  en metros. Para la evaluación de la precisión de la construcción de los mapas bidimensionales, el robot virtual capturó 2 imágenes desde 5 perspectivas

**Tabla 3.** Parámetros principales de los experimentos para la ubicación en el mapa y resultado de las evaluaciones de poses para cada experimento.

No.	Mapa	(x,y) m	1	2	3	4	5
1	1	(3.5,0.5)	(3.7,1.7)	(3.6,1.6)	(3.7,1.7)	(3.8,1.8)	(3.3,1.7)
2	1	(0,0)	(0.1,0.2)	(0.2,0.2)	(0.3,0.1)	(0.3,0.0)	(0.3,0.2)
3	2	(0.5,3.5)	(0.0,2.0)	(0.0,2.0)	(0.2,2.2)	(0.0,2.0)	(0.2,2.2)
4	3	(4,4)	(4.0,3.8)	(3.9,3.9)	(4,4)	(3.9,3.9)	(3.9,3.9)

diferentes desde las 4 posiciones a evaluar de las paredes más cercanas. Ejemplos de las capturas realizadas por el robot se observan en la figura 7, para cada posición se capturaron dos imágenes correspondientes a las paredes más cercanas. Las dos primeras imágenes pertenecen a la posición (0,0), mientras que las otras dos pertenecen a la posición (4,4) de la habitación. Los resultados se muestran en la Tabla 3, se muestra las cinco evaluaciones con dos imágenes cada una para cada experimento. Para los cuatro experimentos del par de imágenes por evaluación, el módulo entregó las respectivas poses plasmadas en cada columna. Las poses obtenidas son cercanas a las reales, algunas llegan a ser casi precisas.



**Fig. 7.** Ejemplos de capturas realizadas por el robot NAO virtual. Dos primeras imágenes, posición (0,0), dos últimas posición (4,4). Diferentes perspectivas.

Los resultados obtenidos arrojaron la siguiente precisión: primer experimento  $\pm(0,12, 1,2)$ , segundo experimento  $\pm(0,3, 0,16)$ , tercer experimento  $\pm(0,42, 1,42)$  y cuarto experimento  $\pm(0,06, 0,1)$ . Como se esperaba, el tercer mapa bidimensional es el que tiene mejor precisión. Si se desea tener más precisión en la construcción del mapa bidimensional es necesario tomar más capturas mientras se realiza el aprendizaje de la habitación y considerar un número elevado de neuronas para entrenar ala red neuronal.

#### 4. Conclusión

En este trabajo se ha presentado el desarrollo de un algoritmo para la construcción de mapas bidimensionales mediante odometría inercial y elementos visuales. La construcción del mapa bidimensional se realiza haciendo uso de un módulo de reconocimiento de objetos presentado en [1], basado en características locales y redes neuronales artificiales no supervisadas. Este módulo se utiliza para aprender la habitación y asociarle una pose a cada neurona que comprende la

red entrenada para la representación del mapa bidimensional. Los experimentos plasmados en este trabajo se realizaron mediante simulación en Webots y con un robot NAO virtual. Al trabajar con imágenes provenientes de un robot virtual, además un ambiente virtual, estas disminuyen considerablemente su calidad. Sin embargo, los resultados son aceptables ya que se logró la construcción de un mapa bidimensional de la habitación y se realizaron experimentos de ubicación con una precisión de hasta  $\pm(0,06, 0,1)$ . Se considera que estos resultados pueden mejorar en un ambiente real y con una plataforma robótica real ya que la calidad de las imágenes es un punto clave para la experimentación. La construcción del mapa bidimensional tiene como trabajos futuros el uso de SLAM visual para mejorar la integración de los datos provenientes de la odometría inercial y los elementos visuales.

## Referencias

1. Flores, K.L., Trujillo-Romero, F.: Free form object recognition module using A-KAZE and GCS. In: CORE 2016 - International Congress on Computer Science (2016)
2. WEBOTS: <https://www.cyberbotics.com/> (2018)
3. NAO: Software 1.14.5 documentation, <http://www.doc.aldebaran.com> (2018)
4. International Federation of Robotics: <https://ifr.org> (2018)
5. Robocup@Home: <http://www.robocupathome.org> (2018)
6. Song, P., Zhang, L., Xiao, J.: Robot in a room: toward perfect object recognition in closed environments. In: Computer Vision and Pattern Recognition, Cornell University Library, <http://arxiv.org/abs/1507.02703> (2015)
7. Sasaki, H., Kubota, N., Sekiyama, K., Fukuda, T.: Multiple object detection for intelligent robot vision by using growing neural gas. In: International Symposium on Micro-NanoMechatronics and Human Science, IEEE, pp. 80–85 (2009)
8. Krause, E., Zillich, M., Williams, T., Scheutz M.: Learning to recognize novel objects in one shot through human-robot interactions in natural language dialogues. In: AAAI Publications: Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
9. Yang, Y., Li, Y., Fermler, C., Aloimonos, Y.: Robot learning manipulation action plans by watching unconstrained videos from the world wide web. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
10. Xi, W., Ou, Y., Peng, J., Yu, G.: A new method for indoor low-cost mobile robot SLAM. In: IEEE International Conference on Information and Automation (ICIA), pp. 1012–1017 (2017)
11. Panzieri, S., Pascucci, F., Setola, R., Ulivi, G.: A low cost vision based localization system for mobile robots. [online]. <https://www.researchgate.net/publication/244958230> (2018)
12. Trevor, T.: Mapping of indoor environments by robots using low-cost vision sensors. Queensland University of Technology (2009)
13. Munguia-Alcalá, R. F., Grau-Saldes, A.: SLAM con mediciones angulares: método por triangulación estocástica. Ingeniería en investigación y tecnología, 14(2), 257–274 (2013)
14. Ibarra-Zannatha, J.M., Hernández, E., Cisneros, R., Lavín, J.E., Neira, J.: Desarrollo de un sistema slam visual con reconstrucción 3D monocular de marcas

- orientadas para un humanoide. *Sistemas, Cibernética e Informática*, 6(2), 12–22 (2009)
15. Shuhuan, W., Kamal, M. O., Ahmad, B. R., Yixuan, Z., Yongsheng, Z.: Indoor SLAM Using Laser and Camera with Closed-Loop Controller for NAO Humanoid Robot: *Abstract and Applied Analysis*, 8 (2014)
  16. Yan, W., Weber, C., Wermter, S.: Learning indoor robot navigation using visual and sensorimotor map information. *Frontiers in Neurorobotics*, 15 (2013)
  17. Tjernberg, I.: Indoor Visual Localization of the NAO Platform. Master's Thesis at CSC. KTH Datavetenskap och kommunikation (2013)
  18. Hernández, E.: SLAM Visual para un Robot Humanoide. Tesis de Maestría en Ciencias en Control Automático, Cinvestav (2012)
  19. Alcantarilla, P. F., Nuevo, J., Bartoli, A.: Fast explicit diffusion for accelerated features in nonlinear scale spaces. In: *British Machine Vision Conference (BMVC)* (2013)
  20. Alcantarilla, P. F., Bartoli A., Davison A. J.: Kaze features. In: *ECCV*, Springer-Verlag Berlin Heidelberg, pp. 214–227 (2012)
  21. Fritzke, B.: Growing cell structures, a self-organizing network for unsupervised and supervised learning. *Neural Network*, 9, pp. 1441–1460 (2016)