

# Inteligencia Artificial

## Proyecto: Localización de robots

Verónica Esther Arriola Ríos  
Pedro Rodríguez Zarazúa  
Luis Alfredo Lizárraga Santos

Fecha de entrega: Miércoles 1 de Junio de 2016

### 1. Objetivo

Conocer e implementar una técnica de localización utilizada en aplicaciones reales, específicamente para los robots *Rhino* y *Minerva* en los museos *Deutsches Museum Bonn* y *National Museum of American History*, respectivamente.

### 2. Introducción

La localización robótica ha sido considerado como uno de los problemas fundamentales de la robótica. El objetivo de la localización es lograr estimar la posición del robot en un ambiente, apoyándose en un mapa de este y con lecturas de sensores.

Las técnicas que se han desarrollado hasta la fecha tratan de resolver uno de los dos tipos de localización que hay:

- *Localización local o rastreo.* Estas técnicas tratan de compensar errores odométricos durante el movimiento del robot. Requieren tener una aproximación de la posición del robot en el entorno todo el tiempo, ya que si la pierden es casi imposible volver a recuperar la posición.
- *Localización global.* Estas técnicas están diseñadas para estimar la posición del robot globalmente, es decir, no es necesario tener un aproximado de su posición. Estas técnicas pueden resolver el problema de localizar al robot al momento de encenderlo, al igual que permiten que se lleve al robot a una posición aleatoria del entorno durante su operación y recuperar su posición.

Como se podrán dar cuenta, las técnicas de localización global son más poderosas que las locales.

Para esta práctica desarrollaremos la técnica de localización global de Markov.

### 3. Localización de Markov

La localización de Markov utiliza un sistema probabilístico que mantiene una densidad de probabilidad de la posición del robot sobre todo el entorno. Esta densidad tiene muchas maneras de representar la información sobre la posición del robot. Por ejemplo, el robot puede iniciar con una distribución de probabilidad uniforme representando que no tiene idea de dónde se encuentra en el entorno. En el caso en el que el robot está muy seguro de su posición, la distribución de probabilidad se convierte en una distribución unimodal centrada en la posición del robot.

Veremos un ejemplo sencillo para ilustrar los conceptos de la localización de Markov

#### 3.1. Ejemplo

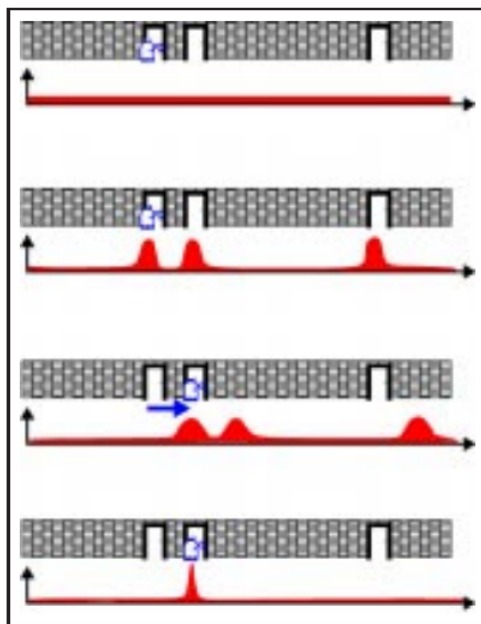


Figura 1: La idea básica de localización de Markov, entorno.

Consideremos el entorno mostrado en la figura 1. Para simplificar, asumamos que el robot sólo puede moverse en una dimensión (enfrente-atrás).

Ahora, supongamos que posicionamos el robot en algún lugar aleatorio del entorno, pero no le informamos al robot cuál es su posición. La localización de Markov representa este estado de “confusión” como una distribución de probabilidad uniforme sobre todo el conjunto de posibles posiciones del entorno, como lo muestra la primer gráfica de la figura 1. Asumamos que el robot hace una medición con sus sensores y determina que está al lado de una puerta. La localización de Markov modifica la distribución de probabilidad de tal manera que las posiciones que se encuentran a un lado de puertas tengan mayor probabilidad, esto queda ilustrado en la segunda gráfica de la figura 1. Notemos que la distribución resultante es multimodal, ya que la información obtenida por los sensores es insuficiente para determinar exactamente la posición del robot. También notemos que las posiciones que no se encuentran a un lado de una puerta aún tienen una probabilidad mayor que cero, esto es porque las mediciones de los sensores contienen ruido. Ahora, supongamos que el robot se mueve un metro hacia el frente. La localización de Markov incorpora este movimiento para que la distribución de probabilidad cambie como se representa en la tercer gráfica de la figura 1. Finalmente, asumamos que el robot vuelve a tomar una medición con sus sensores. Incorporando esta información a la obtenida anteriormente, vemos como la distribución de probabilidad cambia (cuarta gráfica de la figura 1) para poder determinar la posición del robot con una muy alta probabilidad, mientras que todas las demás posiciones tienen una probabilidad casi nula.

## 4. Desarrollo

Primero, iniciemos con el contenido y especificación del mundo en el cual se moverá el robot. Este mundo contendrá obstáculos y será cuadrado (imagínense un cuarto o una oficina), las paredes serán obstáculos y cualquier otra cosa que se pueda encontrar en un cuarto u oficina normal: sillas, mesas, escritorios, etc.

Nuestro robot tendrá tres sensores:

1. láser: mide la distancia desde el robot hacia el primer obstáculo en línea recta del sensor.
2. odométrico: mide la distancia recorrida por el robot (en cm).
3. de giro: mide el giro del robot en grados.

Ahora, como el robot podrá moverse en 8 direcciones ( $0^\circ$  a  $360^\circ$  en aumentos de  $45^\circ$ :  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ , ...) deberán tener una representación del mundo por cada dirección de movimiento. El mundo lo pueden representar como un arreglo de dos dimensiones, donde el tamaño de cada celda está dado en centímetros. Una vez que se tiene la representación del mundo, lo

siguiente sería inicializarlo.

Deberán posicionar al robot aleatoriamente en el mundo, después deberán calcular la probabilidad de que el robot se encuentre en alguna celda, llamémosle a esta probabilidad *creencia*. Al inicio esta probabilidad será igual para toda celda:

$$\frac{1}{\#de\ celdas\ que\ no\ son\ obstaculo}$$

Después, deberán calcular la distancia desde cada celda a cada obstáculo en línea recta. Por ejemplo, si su robot tiene 8 sensores y cada uno está ubicado a 45° de separación, tendrán que calcular las distancias en línea recta hacia arriba, abajo, izquierda, derecha y diagonales hasta que se topen con un obstáculo. Esta información la deberán guardar en cada celda para evitar estarla calculando cada que se visita la celda.

Antes de continuar se necesita definir un poco de notación.

## 5. Notación

Definamos la posición del robot usando una variable tridimensional  $l = \langle x, y, \theta \rangle$ , donde  $x$  y  $y$  son las coordenadas del robot y  $\theta$  es la dirección en la que está viendo el robot. Sea  $l_t$  la posición real del robot en el tiempo  $t$ , y  $L_t$  la variable aleatoria que modela la probabilidad de que el robot se encuentre en una posición  $l$  dada. Como el robot no sabe su posición exacta, su *creencia* o  $Bel(L_t)$  es una distribución de probabilidad sobre el espacio de posibles posiciones. Esta *creencia* nos permite saber cual es la probabilidad de que se encuentre en una celda  $l$  en el tiempo  $t$ , formalmente  $Bel(L_t = l)$ .

También aprovecharé para definir a las lecturas de los sensores como:  $s_T$  la lectura del láser,  $\theta_t$  la lectura del giro y  $a_T$  la lectura del sensor odométrico. A estas lecturas se les tendrá que agregar ruido, esto lo lograrán especificando algunas variables:

1.  $\sigma$ : la varianza de la distribución normal utilizada más adelante.
2.  $S$ : porcentaje de ruido de la lectura del odómetro, se usa para calcular  $\sigma_x$  y  $\sigma_y$ .
3.  $\sigma_\theta$ : ruido para la lectura del sensor de giro.

Una vez definida la notación, procederemos a ver el algoritmo completo de esta técnica de localización.

## 6. Algoritmo

El algoritmo es el siguiente:

```

for all posicion l en mundo do
     $Bel(L_0 = l) \leftarrow P(L_0 = l) = \frac{1}{n}$ 
end for
for all posicion l en mundo do
    determinar las distancias hacia obstaculos
end for
while true do
    decidir si se mueve el robot
    if no se movio el robot then
         $\alpha_T \leftarrow 0$ 
        for all  $\theta$  posibles do
            for all posicion l en mundo do
                 $Bel(L_T = l) \leftarrow P(s_T | l) * Bel(L_T - 1 = l)$ 
                donde  $P(s_T | l) = \frac{1}{(\sqrt{2\pi}\sigma)} * e^{(\frac{-(s_T - distancia_{real})^2}{2\sigma^2})}$ 

                 $\alpha_T \leftarrow \alpha_T + Bel(L_T = l)$ 
            end for
        end for
        for all  $\theta$  posibles do
            for all posicion l en mundo do ▷ Ahora se normaliza la creencia
                 $Bel(L_T = l) \leftarrow \alpha_T^{-1} * Bel(L_T = l)$ 
            end for
        end for
    end if
    if se movio el robot then
        if se obtuvo  $\theta_T$  then ▷ Se obtuvo una lectura del sensor de giro
            for all  $\theta$  posibles do
                for all posicion l en mundo do
                     $Bel(L_T = l) \leftarrow [\sum_{l', \theta} P(l | l', \theta_T) * Bel(L_T - 1 = l')]$ 
                    donde  $P(l | l', \theta_T) = \frac{1}{(2\pi\sigma_\theta)} * e^{(\frac{[(\theta - \theta_{del\ robot}) - \theta_T]^2}{(\sigma_\theta)^2})}$ 
                end for
            end for
        end if
    end if
    if se obtuvo  $a_T$  then ▷ Se obtuvo una lectura del sensor odométrico
        for all  $\theta$  posibles do
            for all posicion l en mundo do

```

```

Bel(LT = l) ← [∑l',θ P(l | l', aT) * Bel(LT - 1 = l')]
donde P(l | l', aT) =  $\frac{1}{(2\pi\sigma_x\sigma_y)} * e^{-\frac{1}{2}(\frac{(x'+a_T\cos\theta-x)^2}{(\sigma_x)^2} + \frac{(y'+a_T\cos\theta-y)^2}{(\sigma_y)^2})}$ 
y con  $\sigma_x = S * a_T * \cos(\theta)$ ,  $\sigma_y = S * a_T * \sin(\theta)$ 
end for
end for
end if
end if
end while

```

## 7. Desarrollo e implementación

Esto se implementará en processing, recuerden que debe estar bien hecho y comentado, ya que este proyecto vale el 20 % de su calificación.

Su aplicación deberá mostrar el mundo, los obstáculos y la probabilidad de cada celda usando distintos colores.

## 8. Notas adicionales.

La práctica es individual, anexen a su código un archivo `readme.txt` con su nombre completo, número de cuenta, número de la práctica y cualquier observación o notas adicionales (posibles errores, complicaciones, opiniones, críticas de la práctica o del laboratorio, cualquier comentario relativo a la práctica).

Compriman la práctica en un solo archivo (.zip, .rar, .tar.gz) con la siguiente estructura:

- ApellidopaternoNombreProyecto.zip (por ejemplo: LizarragaLuisProyecto.zip)
  - ApellidopaternoNombreProyecto
    - Proyecto
      - ◇ Localizacion.java
  - readme.txt

La práctica se entregará en la página del curso en la plataforma AVE Ciencias.  
O por medio de correo electrónico a [luislizarraga@ciencias.unam.mx](mailto:luislizarraga@ciencias.unam.mx) con asunto Proyecto[IA 2016-2]

**La fecha de entrega es hasta el día miércoles 1 de Junio a las 23:59:59 hrs.**

## Referencias

- [1] Dieter Fox, Wolfram Burgard et. al. *Markov Localization for Mobile Robots in Dynamic Environments*, 1999 : Journal of Artificial Intelligence .