

## Navegación de Robots Móviles a través de FLOOD-FILL

J. Sandoval<sup>1</sup>, J.J. Medel<sup>1,2</sup>

<sup>1</sup>Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada Unidad Legaria del Instituto Politécnico Nacional, Legaria 694. Colonia Irrigación, 11500 México D. F.

<sup>2</sup>Centro de Investigación en Computación del Instituto Politécnico Nacional, Av. Juan de Dios Bátiz s/n esq. Miguel Othón de Mendizábal Unidad Profesional "Adolfo López Mateos". Colonia Industrial Vallejo, 07738 México D. F.

### Resumen

Se describe el estudio de la técnica flood-fill, para obtener un algoritmo en la toma de decisiones sobre la navegación autónoma en robots móviles posicionados en un ambiente continuo que se convierte en discreta por medio de una rejilla de posiciones, evitando colisiones y optimizando la distancia de recorrido.

### Introducción

El robot móvil considerado, tiene una posición  $(x_j, y_j)$ , así como los obstáculos  $(x_k, y_k)$  también se encuentran sobre un plano de coordenadas  $(x_i, y_i)$  con valores discretos. La problemática por la cual se busca navegar de forma discreta en un ambiente continuo, se basa en evitar errores de localización, posición y locomoción, en este sentido, la propuesta de navegación se enfoca en un plano cuadrículado en forma simétrica que permita mantener un perfil discreto en los movimientos del robot móvil [2]. Siendo un plano finito, existe una región libre de obstáculos por la cual, el robot móvil puede navegar de forma correcta sin colisionar, por tanto un plano  $P$  con dimensiones  $(x_i, y_i)$  tiene la ruta óptima hasta  $P_{ij}^{(m)}$ . Respecto a la técnica Flood-Fill (FF) se basa en un comportamiento de incremento o retroceso válido para ambiente discreto mediante una unidad espacial adyacente con las mismas propiedades [1].

### Procedimiento experimental

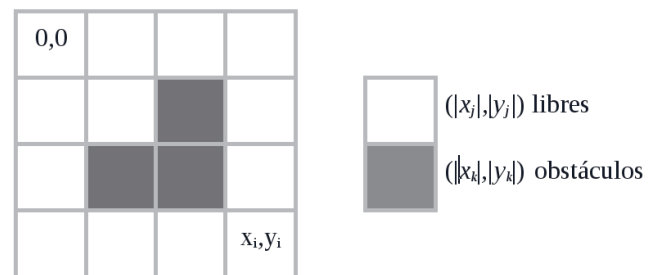
Dado un conjunto de posiciones libres en un ambiente acotado, solamente podrán ser rutas válidas las que satisfagan las siguientes sentencias.

$$\{(x_j, y_j) \&\&(x_{j+1}, y_j)\}, \{(x_j, y_j) \&\&(x_{j-1}, y_j)\} \\ \{(x_j, y_j) \&\&(x_j, y_{j+1})\} \text{ y } \{(x_j, y_j) \&\&(x_j, y_{j-1})\}$$

De tal forma, que sean adyacentes en forma horizontal y/o vertical, debido a la configuración propuesta por la locomoción del robot móvil en cuestión [2], así mismo se considera una región completa cuando se obtengan las fronteras de la ruta libre. En forma iterativa para la fig. 1

```
for (int j = 0; j < 5; j++)
{
    for (int i = 0; i < 5; i++)
    {
         $(x_j, y_j) \&\&(x_{j+1}, y_j)$ 
    }
}
```

para incrementos de izquierda a derecha y de arriba hacia abajo, sin embargo, resulta igual en cualquiera de las cuatro posibles orientaciones.



**Figura 1.** Representación de posiciones libres y con obstáculos en un plano segmentado por rejillas

Por otro lado, planteando un método recursivo se tiene:

```
// iniciar en cualquier posición
SI  $\{(x_j, y_j) \&\&(x_{j+1}, y_j)\} \{ \{(x_j, y_j) \&\&(x_{j-1}, y_j)\} \\ \{(x_j, y_j) \&\&(x_j, y_{j+1})\} \{ \{(x_j, y_j) \&\&(x_j, y_{j-1})\} \}$ 
ENTONCES  $(x_{j+1})$  //Sucesivamente
... SINO // Sucesivamente
 $(x_j, y_j)$  // Son los límites de una región.
FIN
```

### Resultados y Análisis

En el método iterativo, la reducción en operaciones es de 1 a 10 en comparación con el método recursivo por cada movimiento, en cambio el tiempo de operación para encontrar toda la región válida de la Fig. 1 para el método iterativo es de 23 movimientos contra 12 del método recursivo, siendo una ventaja en navegación para el propósito en el caso de navegación autónoma en busca de un objetivo específico [2].

### Agradecimientos

Agradecemos, por su apoyo, al Instituto Politécnico Nacional, así como al Consejo Nacional de Ciencia y Tecnología.

### Referencias

- [1] M. Kalisiak, M. V. Panne, "RRT-blossom: RRT with a local flood-fill behavior", Proceedings of International Conference on Robotics & Automation 2006, Orlando, Florida
- [2] J. J. Medel, J. Sandoval, "Autonomous Mobile Robot to Exit Path Planning for Floor Tile Room" ISPR 09, U.K. 2009.