

Diseño e implementación de una interfaz entre las bibliotecas OMPL y nD FMM

TRABAJO FIN DE GRADO

13 de octubre de 2015

Álvaro Muñoz Serrano

Departamento de Sistemas y Automática



Universidad
Carlos III de Madrid



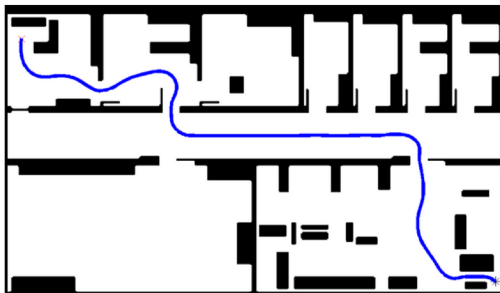
Motivación y objetivos

Marco teórico

Trabajo Realizado

Resultados y conclusiones

Motivación





nD FMM: n-Dimensional Fast Marching Methods

OMPL: Open Motion Planning Library



Crear una interfaz que permita complementar las funcionalidades que ofrecen ambas bibliotecas.

- ▶ nD FMM
 - ▶ Cargar entornos y robots 3D
 - ▶ Aplicar orientación de los robots
 - ▶ Visualizar resultados
- ▶ OMPL
 - ▶ Utilizar algoritmos basados en Fast Marching

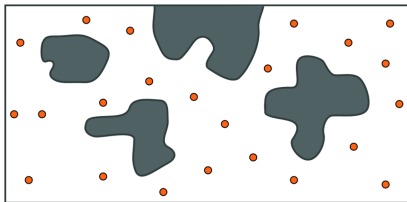


OBJETIVO: Trasladar un objeto de un lugar a otro sin colisionar con obstáculos.

- ▶ Métodos combinatorios
- ▶ Métodos basados en muestreo aleatorio

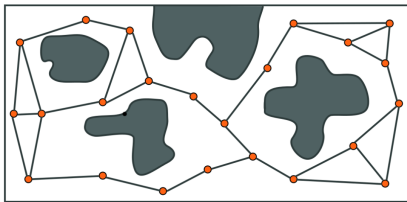
OBJETIVO: Trasladar un objeto de un lugar a otro sin colisionar con obstáculos.

- ▶ Métodos combinatorios
- ▶ Métodos basados en muestreo aleatorio



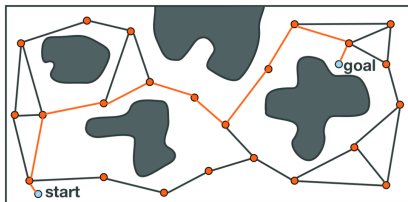
OBJETIVO: Trasladar un objeto de un lugar a otro sin colisionar con obstáculos.

- ▶ Métodos combinatorios
- ▶ Métodos basados en muestreo aleatorio



OBJETIVO: Trasladar un objeto de un lugar a otro sin colisionar con obstáculos.

- ▶ Métodos combinatorios
- ▶ Métodos basados en muestreo aleatorio

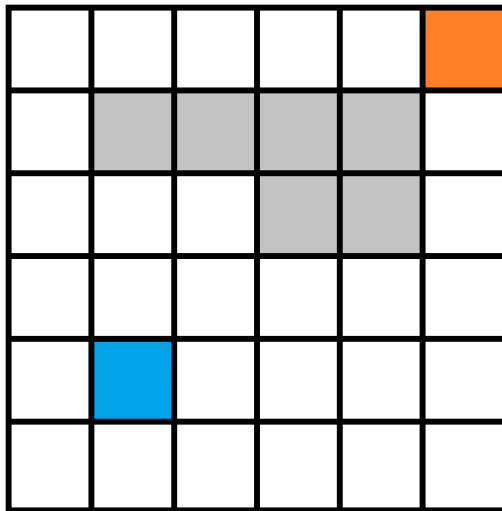




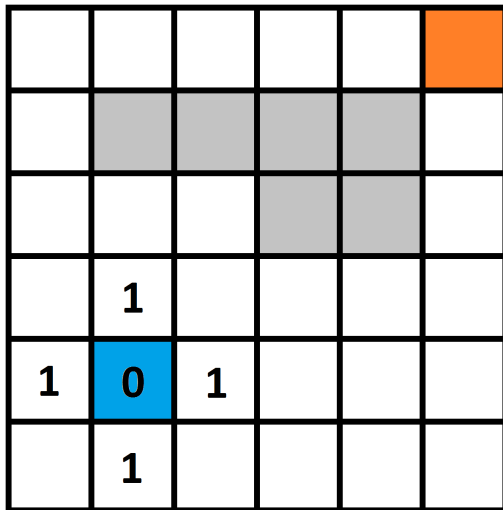
- ▶ El FMM es un método numérico que modeliza el comportamiento de una onda.
- ▶ Se basa en la ecuación Eikonal

$$1 = F(x) |\nabla T(x)| \quad (1)$$

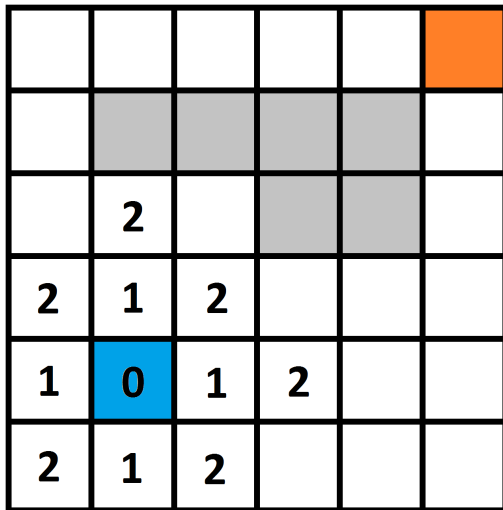
Fast Marching Method



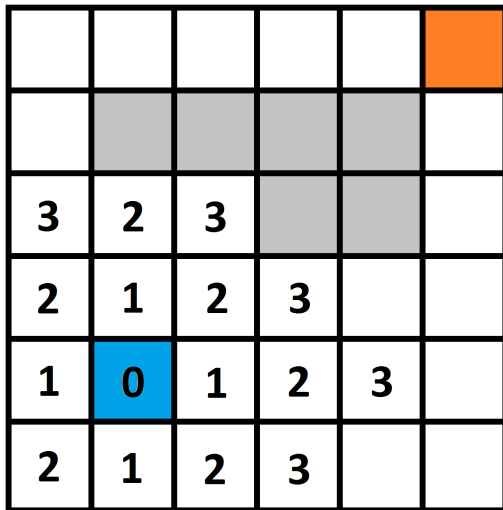
Fast Marching Method



Fast Marching Method



Fast Marching Method



Fast Marching Method



5					
4					
3	2	3			
2	1	2	3	4	5
1	0	1	2	3	4
2	1	2	3	4	5

Fast Marching Method



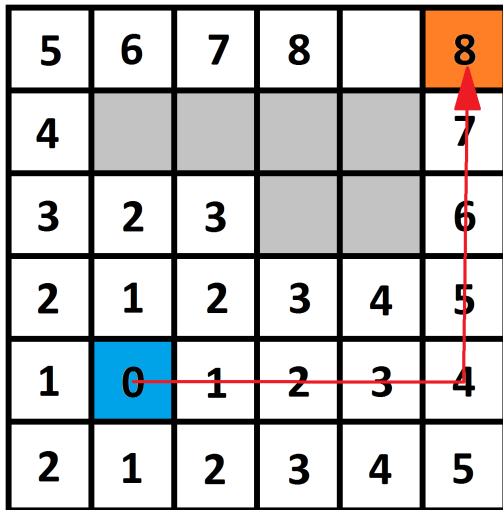
5	6	7			
4					7
3	2	3			6
2	1	2	3	4	5
1	0	1	2	3	4
2	1	2	3	4	5

Fast Marching Method



5	6	7	8		8
4					7
3	2	3			6
2	1	2	3	4	5
1	0	1	2	3	4
2	1	2	3	4	5

Fast Marching Method

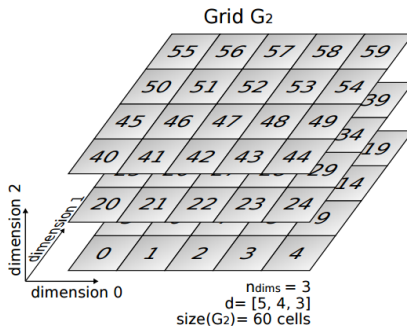
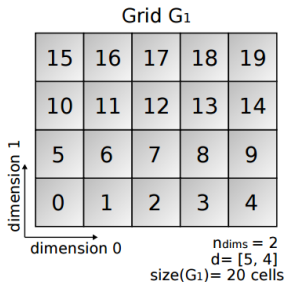


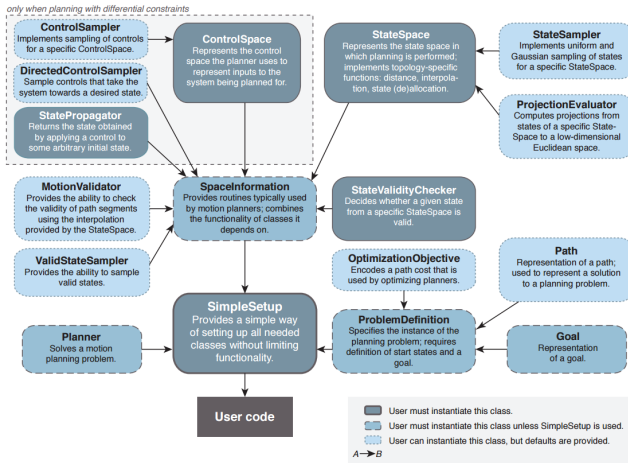


Clases principales de nD FMM

- ▶ nDGridMap: Contiene la información del espacio.
- ▶ Solver: Implementa los algoritmos basados en FMM para la expansión de la onda.
- ▶ GradientDescent: Calcula el camino mediante la aplicación del descenso de gradiente.

Representación del espacio en nD FMM



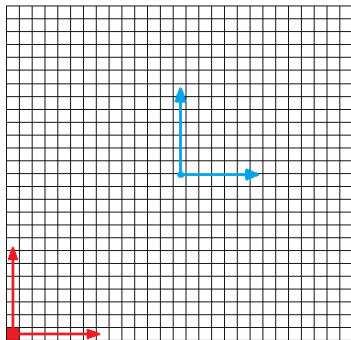




- ▶ Describe el espacio de la manera mas general posible
- ▶ Encapsulados en clase StateSpace
 - ▶ RealVectorStateSpace
 - ▶ SO2StateSpace
 - ▶ SO3StateSpace
 - ▶ CompoundStateSpace
 - ▶ SE2StateSpace
 - ▶ SE3StateSpace
- ▶ Los espacios de estados son continuos

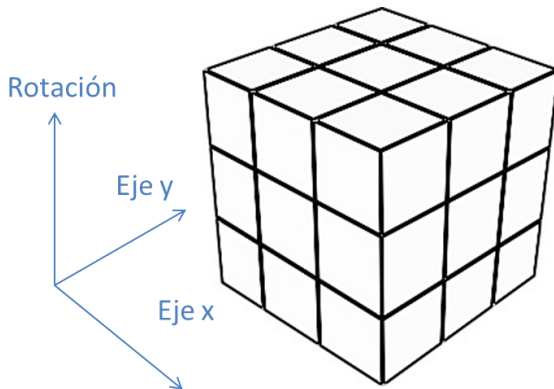


- ▶ Expansión de la OMPL para la resolución de problemas SE(2) y SE(3)
- ▶ SE2RigidBodyPlanning y SE3RigidBodyPlanning
- ▶ Implementa un detector de colisiones
- ▶ Permite cargar archivos COLLADA de entornos y robots en 3D
- ▶ Incluye una interfaz gráfica de usuario



1. Utilizar método $as<T>$ para hacer casting de StateSpace
2. Obtener vector de coordenadas
3. Obtener límites del entorno
4. Unificar ejes de coordenadas
5. Convertir coordenadas a índice en la rejilla

Conversión de la rotación



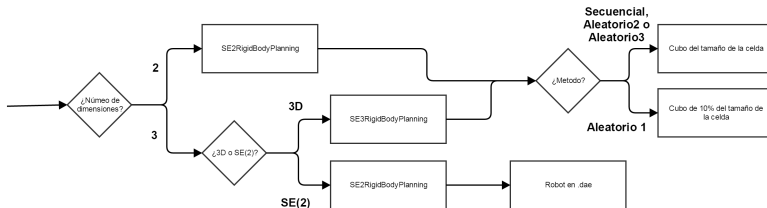


- ▶ No todos los algoritmos requieren mapas exactos
- ▶ Es preferible en algunos casos minimizar el coste computacional
- ▶ Dos tipos de métodos para convertir mapas
 - ▶ Muestreo secuencial
 - ▶ Muestreo aleatorio

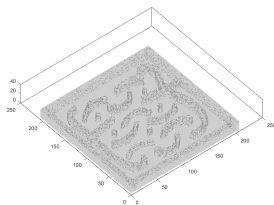
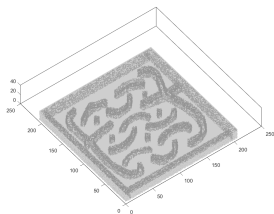
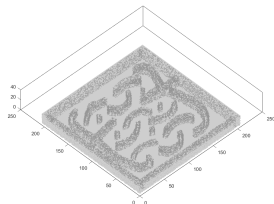
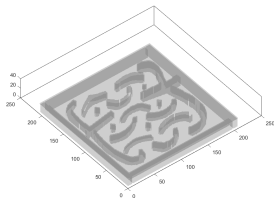
Métodos aleatorios



16



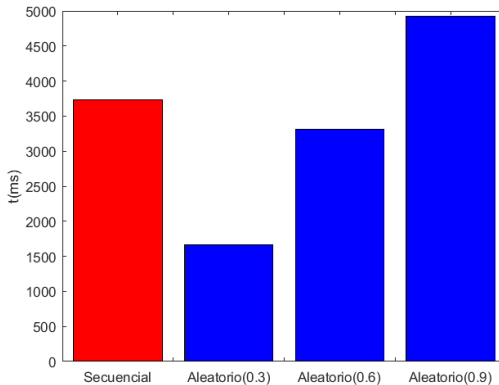
Comparación de métodos 1 y 2



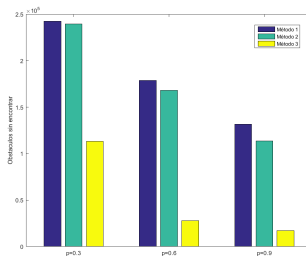
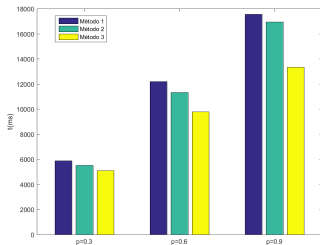
Comparación de métodos 1 y 2



18



Comparación de métodos 1 y 2



► MapLoader

- Cargar mapas en 3D y convertirlos a nDGridMap
- Representar la rotación en la rejilla
- Convertir camino de GradientDescent a *PathGeometric* de OMPL
- Exportar soluciones al formato de OMPL.app

► nDFMM

- Encapsular algoritmos nD FMM
- Instanciar automáticamente todo lo necesario para resolver el problema
- Convertir solución a formato OMPL



Video example



¿Alguna pregunta?



Universidad
Carlos III de Madrid