

BEEBOTS, ESTRUCTURA DE DATOS PARA PREDECIR COLISIONES

Santiago Hincapié Murillo
Universidad EAFIT
Medellín, Colombia
shincapiem@eafit.edu.co

Santiago Valencia Arango
Universidad EAFIT
Medellín, Colombia
svalenciaa@eafit.edu.co

Andrés Almanzar Restrepo
Universidad EAFIT
Medellín, Colombia
aalmanzarr@eafit.edu.co

Mauricio Toro
Universidad EAFIT
Medellín, Colombia
mtorobe@eafit.edu.co

RESUMEN

Se tiene como objetivo poder evitar la colisión entre diferentes objetos. Para hacerlo tomaremos como objetos abejas robóticas. La razón de esta elección se debe a una simulación siendo las abejas robóticas una posible solución a la colisión entre dos o más objetos.

La magnitud que recae sobre este problema es de vital importancia porque las abejas son unos de los principales actores en la polinización.

Es de suma importancia encontrar una solución puesto que, con el paso del tiempo, crece la demanda de alimentos, pero como respuesta al aumento de la población se tiene mayor uso de pesticidas, incremento de la contaminación, cambio climático, etc. Las abejas se ven afectadas y con ello la producción de alimentos también.

A lo largo de este informe, se expondrán distintos problemas algorítmicos, de baja complejidad y alta velocidad. El objetivo de estos problemas algorítmicos será su aplicación para evitar colisión entre objetos en la vida real.

Palabras clave

Organización de sistemas informáticos → Sistemas embebidos → Estructuras de Datos → Algoritmos → Robótica → Notación O → Complejidad

Palabras clave de la clasificación de la ACM

Computer systems organization → Embedded and cyber-physical systems → Robotic

1. INTRODUCCIÓN

Un factor imprescindible en la polinización es la presencia de las abejas. Desafortunadamente el humano y su constante cambio en la naturaleza ha provocado que las abejas estén en peligro de extinción.

Organizaciones como FAO advierten que hay 100 especies de cultivos que proporcionan el 90% de

alimentos a nivel mundial y de ese centenar unas 71 especies son polinizadas por abejas.

El año 2012 se registró como el peor año de la historia para la apicultura. Esto según estadísticas en Estados

Unidos. En Oregón murieron 50,000 abejas a causa de los efectos causados por los pesticidas.

En Europa, la comisión para el control de seguridad alimentaria de la unión europea (EFSA), confirmó una masiva baja en la población de abejas a causa de un fertilizante llamado neonicotinoides.

Estos informes, como muchos otros nos exponen sin lugar a duda la gran pérdida de agentes polinizadores.

Múltiples grupos, organizaciones e instituciones han dado con frecuencia “soluciones” a este problema. ¿Son estas respuestas efectivas?

2. PROBLEMA

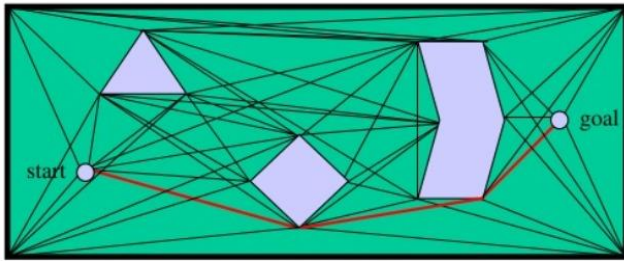
Consideramos útil la implementación de estructuras de bases de datos para la prevención del choque de objetos. Gracias al uso de una estructura de bases de datos podremos consultar la posición y si se está dando alguna colisión entre las abejas robóticas permitiendo así un mayor control de la labor de polinizar cultivos. Ya con el control sobre la colisión de las abejas robóticas, múltiples aplicaciones a futuro serán más fáciles de manejar.

3. TRABAJOS RELACIONADOS

A continuación, se exponen 4 problemas algorítmicos similares.

3.1 Método en espacio cartesiano

Suponiendo que se trabaja en un plano 2D, en el cual el objeto es un punto y los obstáculos son convexos, se elige el camino más corto formado por segmentos que conectan los vértices de los objetos.



Explicación de la gráfica: En el método se definen todos los segmentos que conectan los vértices con “Start”, con “Goal” y entre sí, eliminando las conexiones que intersecan los obstáculos.

En negro los caminos posibles, en rojo el más corto.

Notamos que posee cierto parecido con el algoritmo Dijkstra.

3.2 Colisiones 2D en videojuegos

Uno de los problemas más grande a la hora de la creación de un videojuego, es la colisión. Sin colisiones existirían muchos problemas como: atravesar edificios sin restricción, 2 o más personajes no se chocarían, en videojuegos de shooter las balas no harían daño, en videojuegos de carreras no habría choques, etc.

La manera más fácil de detectar colisiones en videojuegos es simplemente envolver los objetos móviles en simples figuras geométricas, ya sea un cuadrado o un círculo.

Colisión rectángulo-rectángulo:

Para que un rectángulo esté colisionando con otro deben cumplirse cuatro cosas, definiendo dos rectángulos como r1 y r2 diremos que colisionan si:

Lado derecho de r1 es mayor que lado izquierdo de r2

Lado izquierdo de r1 es menor que lado derecho de r2

Lado superior de r1 es mayor que lado inferior de r2

Lado inferior de r1 es menor que lado superior de r2

Si se cumplen estas cuatro condiciones es que ambos rectángulos están colisionando.

Colisión círculo-círculo:

Si la suma de sus radios es mayor que la distancia entre sus centros entonces existe colisión. Así que básicamente hay que hacer dos cosas: Calcular la distancia entre sus centros (que es la distancia entre dos puntos) y comprobar si es menor que la suma de sus radios.

¿Cuál es más rápido?

Simple. El cuadrado debido a que su operación es más simple que la del círculo, debido a que la del círculo contiene raíces cuadradas. Pero los cuadrados tienen una

desventaja frente a los círculos, debido a que los círculos admiten rotaciones. Ya que por mucho que rote un círculo siempre será un círculo, pero en el momento que rote un rectángulo, pierde sus propiedades y por consecuente los cálculos simples no funcionan. En conclusión, es mejor utilizar los rectángulos en situaciones que no se vaya a ejecutar una rotación.

3.3 Búsqueda Heurística en el espacio cartesiano

Este método consta en establecer una función de evaluación que se tenga el objeto en estado x (posición/orientación), se da por:

$$f(x) = g(x) + h(x)$$

siendo:

x el estado (posición/orientación del robot.

f la estimación del coste del camino desde el estado inicial pasando por x hasta el estado final.

g es coste del camino desde el estado inicial a x.

h la estimación del coste del camino desde x al estado final.

Tomado de: Aníbal Ollero Baturone.

ROBÓTICA Manipuladores y robots móviles.

Marcombo; Edición: 1 (8 de julio de 2005).

3.4 Teoría de colisiones 2D: QuadTree

QuadTree es un tipo de estructura de datos representado por nodos, en el cual de cada nodo se desprende 4 nodos hijos. Con los 4 nodos hijos se divide el cuadrante donde estén los objetos que podrían colisionar. Dado el caso que en un cuadrante haya demasiados elementos que puedan colisionar, será necesario que ese cuadrante se vuelva a dividir en 4 partes. Esto se hará hasta que haya un número moderado y manejable de objetos.

Resumen hecho de:

<http://www.dccia.ua.es/dccia/inf/asignaturas/RG/2002/trabajos/colisiones.pdf>

REFERENCIAS

1. ¿Qué pasaría si desaparecen las abejas?
<https://www.sostenibilidad.com/medio-ambiente/que-pasaria-si-desaparecen-las-abejas/>
2. Científicos polacos crean la primera abeja robótica que poliniza como una real.
<http://www.sinembargo.mx/28-11-2016/3120015>
3. Abejas robóticas polinizadoras: ¿realmente son una buena idea?
<https://www.ecologiaverde.com/abejas-roboticas-polinizadoras-realmente-son-una-buena-idea-591.html>
4. Teoría de colisiones 2D: Conceptos básicos
<https://www.genbetadev.com/programacion-de-videojuegos/teoria-de-colisiones-2d-conceptos-basicos>
5. Colisiones 2D en los videojuegos
<http://www.dccia.ua.es/dccia/inf/asignaturas/RG/2002/trabajos/colisiones.pdf>
6. Aníbal Ollero Baturone. ROBÓTICA Manipuladores y robots móviles. Marcombo; Edición: 1 (8 de julio de 2005).