

CONTROL Y SUPERVISIÓN ALGORÍTMICA PARA LA PREVENCIÓN DE COLISIONES ENTRE ABEJAS ROBÓTICAS

Hamilton Smith Gómez Osorio
Universidad EAFIT
Colombia
hsgomez@eafit.edu.co

Santiago Isaza Cadavid
Universidad EAFIT
Colombia
sisazac@eafit.edu.co

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

RESUMEN

Las abejas robóticas son el futuro de la agricultura en su proceso de producción por lo que se hace necesario tener un control y supervisión de la ubicación de estas para evitar colisiones y así lograr el funcionamiento óptimo de las mismas. Estos problemas de colisiones también se presentan en la industria como videojuegos, finanzas entre otros.

Palabras clave: Sorting and Searching, Data structures, Hashing, Data Management.

Sistema de clasificación ACM:

CCS → Theory of computation → Design and analysis of algorithms → Data structures design analysis → Sorting and searching.

INTRODUCCIÓN

Frente a la disminución de la población de abejas que se presenta actualmente y la importancia de estas en el proceso de polinización del sector agrícola se ve un riesgo futuro en los cultivos, por lo que es necesario encontrar una solución a dicho problema. Así nace la idea de crear abejas robóticas las cuales ayuden en este proceso y, para supervisar y controlar su funcionamiento, implementar una estructura de datos en la que por medio de un algoritmo se prevenga la colisión de las mismas.

2. PROBLEMA

Las abejas robóticas implementadas en la agricultura para el proceso de polinización pueden colisionar si están a menos de 100 metros de distancia de otras abejas por lo que es importante solucionar dicho problema para tener un funcionamiento óptimo y una mejora en los procesos.

3. TRABAJOS RELACIONADOS

Encontramos trabajos relacionados con la radiofrecuencia, algunos utilizados en el área de la computación gráfica, otros utilizados en el sector de la privacidad y el manejo de datos, todos relacionados con la colisión, en los que se intentaban prevenir o aprovecharse de ella para solucionar problemas mayores.

3.1 Sistema de identificación por radiofrecuencia (RFID)¹

Cuando hay varias etiquetas (usadas para almacenar la información) y lectores (lee, cambia y verifica la información de la etiqueta) en el mismo canal y transmisión de señal, se genera un problema de colisión debido a las interferencias mutuas entre las etiquetas y los lectores.

Solución: Un algoritmo anticolisión basado en una matriz y esquema de codificación.

Se establecen los datos decodificados en una matriz y luego el lector se encarga de procesar dichos datos por filas, se analizan por parejas y al encontrar una colisión se reemplaza por valor uno (1), en caso contrario se establece un cero (0). Luego de reemplazar las filas, se extraen las colisiones y se siguen analizando las siguientes filas hasta terminar

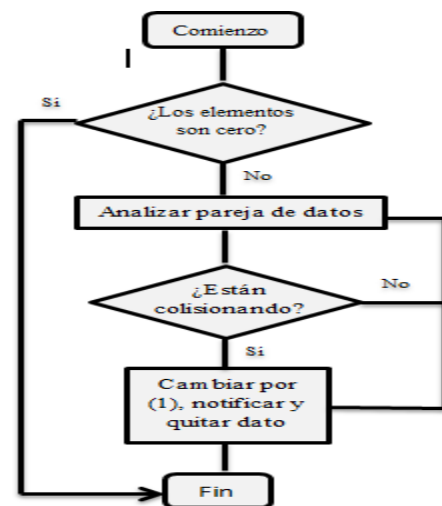


Figura 1: diagrama de flujo RFID

3.2 Volúmenes acotados para detectar colisiones ²

Muy utilizado en la computación gráfica a la hora de realizar videojuegos. Se basa en utilizar formas geométricas básicas encapsulando figuras complejas y utilizando la intersección de estas para determinar cuándo alguna colisiona con la otra; por medio de estas figuras se tiene control de los objetos cuando hay movimientos o cambios de perspectiva, de los más utilizados está AABB y OBB.

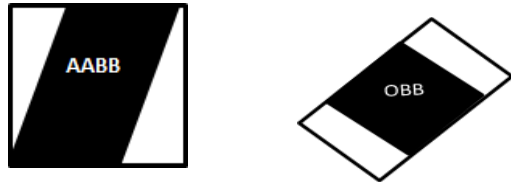


Figura 2: tipos de figuras de acotamiento

3.3 Árbol Octree ³

Para analizar colisiones entre múltiples elementos en un plano se utiliza este método el cual busca subdividir el espacio delimitado en rectángulos de igual medida y luego volver a dividir este último hasta tener una zona relativamente menor en la que sea más fácil y rápido la comparación entre las posiciones de un objeto y otro, teniendo en cuenta su volumen y sus coordenadas.

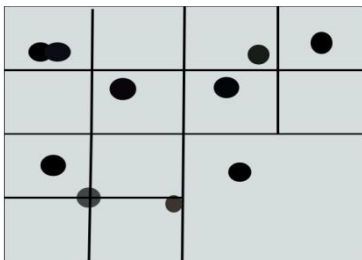


Figura 3: subdivisiones del árbol Octree

3.4 Estructura de datos Spatial Hushing ⁴

Consiste en dividir una zona en cubos con una medida específica, teniendo en cuenta el máximo y mínimo valor de coordenadas, y luego organizar los objetos que estén inscritos dentro de estos en una lista de referencia al índice de la caja en la que se encuentra.

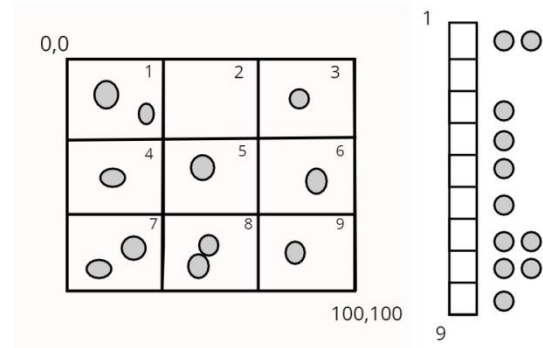


Figura 4: subdivisión y clasificación por cajas

4. Estructura a implementar: Spatial Hushing

4.2 Criterios de diseño de la estructura de datos

Esta estructura es mucho más eficiente en comparación a las demás presentadas ya que desde la construcción del algoritmo se puede definir el tamaño de cada celda y así no necesita analizar ninguna posición de una abeja respecto otra, pues el solo hecho de pertenecer a la misma celda de subdivisión ya representa un riesgo de colisión. Por lo que con esta estructura se logra ahorrar espacio de memoria, evitando cualquier tipo de clasificación y consulta, además de hacer al algoritmo más rápido.

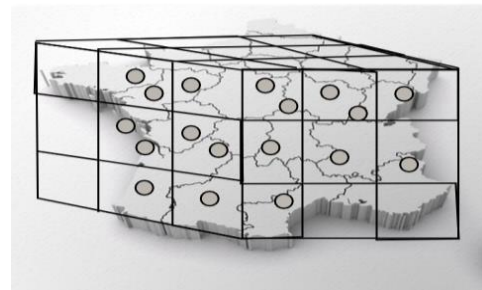


Figura 5: mapa acotado por cajas

Planteamiento de implementación

En el lenguaje de programación Java buscamos implementar una solución óptima con la estructura de datos Spatial Hushing la cual por medio de cubos de volumen 100 m^3 , delimitando la extensión territorial del municipio de Bello, inscribir a las abejas robóticas ubicadas allí, clasificarlas en una lista y posteriormente retornar aquellas que se encuentran en el mismo cubo y reconocer aquellas en riesgo de colisión.

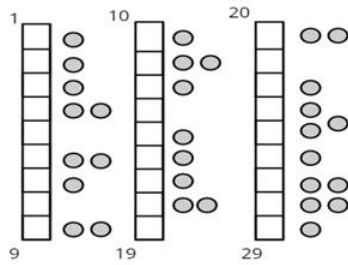


Figura 6: organización de abejas según su posición

AGRADECIMIENTOS

Principalmente agradecemos al proyecto de educación nacional Ser Pilo Paga del Gobierno de Colombia y a la beca Fundación de la Universidad EAFIT por apoyar económicamente nuestro proceso formativo.

Agradecemos también por su asesoría en el proceso de aprendizaje a Daniel Mesa, Monitor del curso de Estructura de Datos y Algoritmos I en la Universidad EAFIT, por sus aportes en clase y su orientación en la realización de este proyecto.

REFERENCIAS

- [1]. Liu, B. and Su, X. An Anti-Collision Algorithm for RFID Based on an Array and Encoding Scheme. Information, 2018, 2078-2489. Accessed August 25, 2018 from Universidad EAFIT: <https://bit.ly/2PEzPhx>
- [2]. Dinas, S. and Bañón J. M. A literature review pf bounding volumes hierarchy focused on collision detection. Ingeniería Competitiva, 2015, 49-62. Accessed August 25, 2018, from Universidad EAFIT: <https://bit.ly/2BMI9sD>
- [3]. Nevala, E. Introduction to Octrees. GameDev.net, 2018. Accessed September 23, 2018: <https://bit.ly/2pxAzJa>
- [4]. Spatial hashing implementation for fast 2D collisions. The mind of Conkerjo, 2013. Accessed September 23, 2018: <https://bit.ly/2xHprNK>
- [4]. How to efficiently remove duplicate collision pairs in spatial hash grid? Stack Overflow, 2015. Accessed September 23, 2018: <https://bit.ly/2O2PPvG>